Visual Tracking in Robotic Minimally Invasive Surgery

Xiaofei Du

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Medical Physics and Biomedical Engineering

University College London

April 18, 2018

I, Xiaofei Du, confirm that the work presented in this thesis is my own.
Where information has been derived from other sources,
I confirm that this has been indicated in the thesis.

Abstract

Intra-operative imaging and robotics are some of the technologies driving forward better and more effective minimally invasive surgical procedures. To advance surgical practice and capabilities further, one of the key requirements for computationally enhanced interventions is to know how instruments and tissues move during the operation. While endoscopic video captures motion, the complex appearance dynamic effects of surgical scenes are challenging for computer vision algorithms to handle with robustness.

Tackling both tissue and instrument motion estimation, this thesis proposes a combined non-rigid surface deformation estimation method to track tissue surfaces robustly and in conditions with poor illumination. For instrument tracking, a keypoint based 2D tracker that relies on the Generalized Hough Transform is developed to initialize a 3D tracker in order to robustly track surgical instruments through long sequences that contain complex motions. To handle appearance changes and occlusion a patch-based adaptive weighting with segmentation and scale tracking framework is developed. It takes a tracking-by-detection approach and a segmentation model is used to assigns weights to template patches in order to suppress background information. The performance of the method is thoroughly evaluated showing that without any offline-training, the tracker works well even in complex environments. Finally, the thesis proposes a novel 2D articulated instrument pose estimation framework, which includes detection-regression fully convolutional network and a multiple instrument parsing component. The framework achieves compelling performance and illustrates interesting properties including transfer between different instrument types and between *ex vivo* and *in vivo* data.

In summary, the thesis advances the state-of-the art in visual tracking for surgical applications for both tissue and instrument motion estimation. It contributes to developing the technological capability of full surgical scene understanding from endoscopic video.

Impact Statement

In various traditional surgical procedures, foreign body material such as staples and clips have to be left within the body of the patient to close the body vessel. To reduce the effect of foreign material, energy-based therapies which heat the tissue to seal the blood vessels and fuse the tissue have been employed. During the energy-based surgery procedures, tissue monitoring reflects the state of tissue, and enables real-time feedback (e.g. multispectral spectrometry) to improve the energy-delivery strategies.

In collaboration with a research team led by Daniel Elson (Hamlyn Centre for Robotic Surgery, Imperial College London), our tracking method that presented in this thesis has been incorporated in a novel surgical system for monitoring of tissue during the surgical procedure. After using multispectral imaging technique to acquire tissue data before and after tissue fusion, our tracking software is used to compensate the artefacts due to respiratory motion and peristalsis. Therefore, functional and structural information on normal and fused tissue can be extracted from the spectral reflectance data. The information is critical for determining whether the tissue fusion was a successful procedure. A U.S. patent has been applied for in respect of a surgical forceps design along with the above system.

Acknowledgements

To my lovely supervisor Dan and my lovely lab fellows, you made my PhD life as colorful as a rainbow and definitely not as scary as I thought.

To my parents and friends, thank you for always supporting me and always being there for me. I feel less lonely when I am by myself, and also because my annoying cat roommate won't leave me alone.

A little something to remind myself: *You cannot find peace by avoiding life, leonard.*

Contents

1 Computer Assisted and Robotic Minimally Invasive Surgery **18**

1.1 Minimally Invasive Surgery 18

1.2 Computer Assisted Interventions in Minimally Invasive Surgery 20

1.2.1 Intra-operative Imaging Techniques 20

1.2.2 Towards Soft Tissue Surgical Navigation 22

1.2.3 Effective Image Guidance 25

1.3 Thesis Structure 26

1.4 Contributions 27

2 Visual Tracking and Applications in Computer Assisted Surgery **29**

2.1 Visual Tracking Framework 29

2.1.1 Visual Feature Representation 30

2.1.2 Observation Model 34

2.1.3 Motion Model 35

2.1.4 Model Adaptor 36

2.2 Computer Vision for Image Guidance in Minimally Invasive Surgery 37

2.2.1 Tissue Surface Tracking and Recovery 37

2.2.2 Surgical Instrument Detection and Tracking 38

2.2.3 Major Challenges for Visual Tracking in Surgical Applications 42

2.3 Discussion 42

3 Non-rigid Deformation Tracking for Soft Tissue Surface **44**

3.1 Introduction 44

3.2 Non-rigid Deformation Tracking 44

3.3 Geometric Mesh Model 45

3.4 Feature-based Tracking 45

3.5 Deformable Lucas-Kanade Method 47

3.6 Template Updating 49

3.7 Experiments and Results 49

3.7.1 Synthetic Data Experiments 49

3.7.2 *In Vivo* Data Experiments 51

3.7.3 Experiments with Multispectral Data 54

3.7.4 Experiments with Tongue Data 56

3.8 Discussion . . . 57

4 Keypoint Based Surgical Instrument Tracking 60
4.1 Introduction . . . 60
4.2 Tracking Pipeline . . . 61
4.3 Generalized Hough Transform for 2D detection . . . 61
4.4 Model Initialization . . . 63
4.5 Histogram-based Segmentation Model . . . 63
4.6 Rotation-invariant Hough Voting Scheme . . . 65
4.7 Model Adaptation . . . 66
4.8 Combining 2D and 3D Tracking . . . 67
4.9 Experiments and Results . . . 68
4.9.1 *Ex Vivo* Experiments . . . 68
4.9.2 *In Vivo* Experiments . . . 74
4.10 Discussion . . . 74

5 Online Tracking-by-Detection of Surgical Instruments 76
5.1 Introduction . . . 76
5.2 Probabilistic Segmentation Model for Patch Weighting . . . 77
5.3 Scale Estimation . . . 79
5.4 Tracking Framework . . . 81
5.5 Experiments and Results . . . 82
5.5.1 Implementation Details . . . 82
5.5.2 OTB Dataset . . . 83
5.5.3 VOT Challenges Datasets . . . 86
5.5.4 Surgical Instrument Tracking . . . 92
5.6 Discussion . . . 98

6 Deep Learning Based 2D Pose Estimation for Articulated Surgical Instruments 102
6.1 Introduction . . . 102
6.2 Model Architecture . . . 104
6.3 Joint Detection and Association Subnetwork . . . 105
6.4 Regression Subnetwork . . . 107
6.5 Multi-instrument Parsing . . . 108
6.6 Experiments and Results . . . 110
6.6.1 Datasets and Analysis . . . 110
6.6.2 *RMIT* Experiments . . . 112
6.6.3 *EndoVis* Experiments . . . 114
6.6.4 *In Vivo* Experiments . . . 118
6.7 Discussion . . . 119

7 Conclusion and Perspectives on Future Research Possibilities **121**
7.1 Overview of Thesis and Scientific Contributions 121
7.2 Challenges and Limitations 122
7.2.1 Tissue Tracking 122
7.2.2 Robust Instrument Tracking 123
7.2.3 Instrument Pose Estimation 123
7.3 Future Research Directions 124

Figures

1.1 Surgeons performing laparoscopic cholecystectomy in a modern operation room with minimally invasive techniques. The use of laparoscope and elongated instruments passing through the access ports give the surgeons access to the internal organ and visualization of the surgical site on a 2monitor. 19

3.1 The left and right images are the template image T and the input image I respectively. F is the set of feature correspondence obtained using feature matching algorithm (shown as cyan). The tissue surface M is modelled as a triangular mesh model (shown as green), so the deformation and motion of the surface M are controlled by the state vector $\mathbf{S}$ consisting of the mesh vertices $\mathbf{v}$. 45

3.2 A hexagonal element h in the undeformed mesh model (shown in green). The distance between co-linear vertices is equal under certain types of hexagon motion. 46

3.3 SCV illumination mapping example: (a) Template frame with ROI (bounding box in yellow); (b) Input frame with the same ROI (bounding box in yellow); (c) The ROI of the input frame is mapped to mimic the illumination condition of the template image using the SCV metric. 48

3.4 Specular highlight removal procedure before tracking: (a) Before highlight removal; (b) Highlight mask; (c) After highlight removal. 50

3.5 Simulation environment experiment setup for generating synthetic image sequences. 50

3.6 Synthetic experiment results: (a) Mean error for ROI 1; (b) The standard deviation of error for ROI 1; (c) Mean error for ROI 2; (d) The standard deviation of error for ROI 2. 51

3.7 Comparison of performance for a FB sequence with camera motion. The first frame (frame 0) and the last frame (frame 50) are identical, so if a ROI is perfectly tracked, it should return to the initial location in the first frame: (a) Frame 0; (b) The DLK frame 50; (c) The PFN frame 50; (d) The PFNLK frame 50. 52

3.8 The comparison of NCC and of tracked point with different tracking methods throughout the FB sequence: (a) NCC between the original template and tracked ROIs; (b) NCC computed after SCV illumination mapping step. 52

3.9 Comparison of performance for occlusion sequence: (Top row) intensity-based DLK method; (Middle row) feature-based PFN method; (Bottom row) our hybrid PFNLK method. 53
3.10 The comparison of the NCC and of tracked point with different tracking methods throughout the occlusion sequence: (a) The NCC between the original template and tracked ROIs; (b) The NCC computed after SCV illumination mapping step; (c) Trajectory of the tracked point; (d) Tracking error of the tracked point. 54
3.11 Multispectral images acquired one wavelength at a time from $\lambda = 480\ nm$ to $\sim 680\ nm$ 55
3.12 The alignment of multispectral images (wavelength $\lambda = 480 \sim 680\ nm$) without and with illumination compensation: (Top row) original DLK method using SSD metric; (Middle row) SCV images; (Bottom row) our modified DLK method using SCV metric. 55
3.13 The original multispectral images and the difference images without and with using SCV metric: (Top row) the template frame ($\lambda = 480nm$) with the tracked ROI and several frames with observable motion; (Middle row) the difference ROI images without compensation; (Bottom row) the difference ROI images with compensation 56
3.14 The original frames and the difference images of another multispectral image sequence. 56
3.15 Vessel misalignment correction. 57
3.16 Tongue tissue motion is compensated by tracking over time in both left and right cameras separately, the registered outputs are used for oxygen saturation (SO_2) and total haemoglobin (THb) estimation: (The 1st and 4th row) tongue tissue patch registration result; (The 2nd and 5th row) SO_2 estimation overlaid on the laparoscopic RGB image; (The 3rd and 6th row) THb estimation overlaid on the laparoscopic RGB image. 58
3.17 Tissue surface tracking is commonly affected by the intrusion of the surgical instruments in restricted surgical procedures. 59

4.1 Challenges for 3D tracking methods: (a) Illumination by other instruments; (b) Out-of-view; (c) Long-term tracking drift. 61
4.2 The left image shows the 2D detection and estimation of the parameters λ_{2D} which are then are used (a) to initialize the 3D parameters λ_{3D}. After the 3D pose is estimated, a new frame is loaded (b) and 2D detection begins again. . . 62
4.3 Shape detection using Generalized Hough Transform 62

4.5 Segmentation model initialization and update strategy: (a) instead of using image region inside the bounding box (red), image region inside the convex hull (green polygon) of the positive keypoints (green circle) is used to initialize and update the foreground histogram. And background histogram is then initialized using pixels from the pixels outside of the surrounding bounding box (blue region). Filled circle with magenta colour indicates the reference centre; (b) foreground probability colourmap illustration, in which blue colour indicates low probability while red colour indicates higher probability; (c) foreground / background classification binary map based on the probability model. 65
4.6 Voting scheme illustration: (a1) keypoints and reference centre on the model (shown in colour); (a2) keypoints and the tracked centre (u, v) on the input frame; in [1], keypoints vote for the reference centre (b1), in the input frame, the rotation θ is estimated by pairwise angular change and vote based on the rotation estimation in (b2); our rotation-invariant voting scheme votes not only for one direction but a circle (c1-c2), in order to improve robustness, keypoint votes for a ring circle, and the rotation θ and scale s are estimated after voting (d1-d2). 67
4.7 Tracking framework illustration. Each row shows one frame tracking example, left column shows the keypoint voting map, middle column shows the convex hull of the positive keypoints, and the right column shows the segmentation map in the search area. 68
4.8 Example frames from our *ex vivo* sequences acquired using a da Vinci® (Intuitive Surgical Inc., CA) classic stereo laparoscope. The images show typical challenges in instrument tracking, such as instrument and tissue based occlusions and sequences where the instrument goes in and out-of-view repeatedly. . 69
4.9 Performance comparison for Dataset I, which contains a tissue occlusion between frames 250-400. 70
4.10 Frame examples of performance comparison between 2D3D tracking and pure 3D tracking under tissue occlusion. (a-d) Pure 3D tracker result; (e-h) 2D3D tracker result. To display the pose more clearly, the overlay is blended with the original frames. 70
4.11 Performance comparison for Dataset II, which contains a instrument occlusion between frames 225-350. 71
4.12 Frame examples of performance comparison between 2D3D tracking and pure 3D tracking under instrument occlusion. (a-d) Pure 3D tracker result; (e-h) 2D3D tracker result. 71
4.13 Performance comparison for Dataset III, which contains out-of-view occlusions between frames 325-350. 72
4.14 Frame examples of performance comparison between 2D3D tracking and pure 3D tracking under out-of-view. (a-d) Pure 3D tracker result; (e-h) 2D3D tracker result. 72
4.15 Performance comparison for the extended tracking sequence, Dataset IV. 73

4.16 Frame examples of performance comparison between 2D3D tracking and pure 3D tracking for long term. (a-d) Pure 3D tracker result; (e-h) 2D3D tracker result. 73
4.17 Frames showing an instrument tracked through an *in vivo* sequence. (a-c) demonstrate good accuracy whereas in (d) a failure mode for our algorithm is exhibited where poor classification on the instrument body causes the 3D tracked to fail to converge correctly. 74

5.1 Example patch weights are shown for the highlighted bounding box displayed in the top corner of the image. The colour bar indicates the weight where 0 is considered more background and 1 is considered to support foreground. 79
5.2 Examples of object undergo challenging transformations for tracking, inclusion of background information or partial object within the bounding box usually degrade the classifier. 79
5.3 Illustration of the scale estimation by using the KLT tracker. Corner Points on the patches are picked in frame $t-1$, and are tracked in the next frame t by the KLT tracker, the distance ratio of point pairs (p^i, p^j) between two frames are used for scale estimation. 80
5.4 Two-level sampling strategy workflow 81
5.5 Comparison of the precision and success plots on the OTB with the top 10 trackers; the PR scores are illustrated with the threshold at 20 pixels and the SR scores with the AUC in the legend. 84
5.6 Comparison of the tracking results of our proposed tracker PAWSS with SOWP [2] and three conventional trackers: TLD [3], SCM [4] and Struck [5] on some especially challenging sequences in the benchmark. 86
5.7 Comparison of the tracking results of our proposed tracker PAWSS with SOWP [2] and three conventional trackers: TLD [3], SCM [4] and Struck [5] on some sequences with scale variations in the benchmark. 87
5.8 The plots for illumination variation, scale variation and occlusion sub-datasets. The number in the title is the number of sequences in that sub-dataset. 87
5.9 The plots for deformation, motion blur and fast motion sub-datasets. 88
5.10 The plots for in-plane rotation, out-of-plane rotation and out-of-view sub-datasets. 88
5.11 The plots for background clutter and low resolution sub-datasets. 89
5.12 The accuracy-robustness score and ranking plots with respect to the baseline and region-noise experiments of VOT2014 dataset. Tracker is better if its result is closer to the top-right corner of the plot. 90
5.13 The expected overlap score ranking plots of VOT2014 dataset. Tracker is better if its result is closer to the right of the plot. 91
5.14 The accuracy-robustness ranking plots and the expected overlap score ranking plot of VOT2015 dataset. Tracker is better if its result is closer to the top-right corner of the plot. The published sota bound is established based on top trackers in recent years. Any tracker with performance over the boundary is considered as a state-of-the-art tracker. 91

5.15 (a) Example frame from each sequence of the *EndoVis* articulated surgical instrument dataset, the last two example image is from test data; (b) The original annotation includes the position of the TrackedPoint, in our annotation, we relabeled the TrackedPoint and also added new annotations for the Head and Shaft points, which are referred as the HeadPoint and the ShaftPoint. 93
5.16 Result example frames from each sequence of the *EndoVis* articulated surgical instrument dataset. The result bounding box and centre point is represented in cyan colour, and the GT centre point is represented in green colour. Scale bar equals 100 pixels. 94
5.17 Tracking accuracy of *EndoVis* Articulated Robotic Surgical Instrument training data under different accuracy threshold with the original and high-quality annotations 95
5.18 Result example frames from each sequence of the *EndoVis* articulated surgical instrument dataset for HeadPoint joint (top row) and ShaftPoint joint (bottom row). The result bounding box and centre point is represented in cyan colour, and the GT centre point is represented in green colour.Scale bar equals 100 pixels. 96
5.19 Accuracy of *EndoVis* Articulated Robotic Surgical Instrument training data under different accuracy threshold with high quality annotation 96
5.20 (a) Example frame from each sequence of the *EndoVis* articulated surgical instrument training dataset; (b) The annotation includes the position of the TrackedPoint. 97
5.21 Result example frames from each test sequence of the *EndoVis* conventional surgical instrument dataset. The result bounding box is represented in cyan colour. Scale bar equals 100 pixels. 98
5.22 Performance comparison of our proposed tracker PAWSS with GHT and two trackers: CST [6] and TLD [3] on (a-c) Dataset I with tissue occlusion, (d-f) Dataset II with instrument occlusion, (g-i) Dataset III with out-of-view occlusion and (j-l) the extended tracking sequence Dataset IV. 99
5.23 Instrument Tracking result with patch weight displayed in the top corner of the image. Scale bar equals 100 pixels. 100

6.1 The pipeline of our proposed pose estimation framework and the detection-regression FCN architectural design. The output of the network is integrated to associate joints and assemble them into the final poses for all instruments in the frame. 104
6.2 The instrument structure is decomposed into N joints and M joint pairs, based on the articulation, instruments for different datasets could have slightly different joint structure. Joints are represented by colour dots, and joint pairs are connected by black lines. (Top) The *EndoWrist* instrument is made up of 5 joints and 4 joint pairs; (Bottom) The Retinal instrument is made up of 4 joints and 3 joint pairs. 105

6.3 Detection subnetwork GT example for Shaft-End joint pair (a): the binary map for Shaft-End pair association map (b), the Shaft (c) and End (d) joint. 106
6.4 Detection subnetwork GT example for Shaft-End joint pair (a): the binary map for Shaft-End pair association map (b), the Shaft (c) and End (d) joint. 107
6.5 Graph relaxing for instrument structure: (a) Fully connected graph; (b) Tree structure graph; (c) A set of bipartite graphs after relaxation, the matching of joint pairs are decided independently. (d) Single joint pair connection example, multiple candidates are detected for each joint type, and the matching is solved by maximum weight bipartite graph matching. 109
6.6 Example frame from each sequence of the single-instrument *RMIT* dataset. . . 110
6.7 The (a) original and (b) original annotation for *EndoVis* dataset, (c) smoke effect simulation and (d) simulation overlaid on the frame. 111
6.8 Example frames from the multi-instrument *in vivo* dataset. 111
6.9 Examples of the single-instrument *RMIT* dataset. The frames are trimmed around the instrument for better visualization. Scale bar equals 50 pixels. . . . 112
6.10 Result example from test set. (a) The original frame; (b) the estimated pose; joint (c1-5) and association (d1-5) score output from detection subnetwork; joint (e1-4) and association (f1-4) output from regression subnetwork. 116
6.11 Examples of original *EndoVis* test data. Our network is able to detect a new instrument (Curved Scissor) which are not seen in the training data. Scale bar equals 100 pixels. 116
6.12 Examples of smoke-simulated *EndoVis* test data. Our network is able to detects instruments which are not seen in the training data even under smoke simulation. Scale bar equals 100 pixels. 117
6.13 Examples of failure cases of *EndoVis* test set. (a) Occuded joints are miss detected; (b) The head joint of the new Curved Scissor instrument on the left is not well localized. Scale bar equals 100 pixels. 119
6.14 Examples of *in vivo* data with our fine-tuned model. The results demonstrate the capacity of our framework to be applied to real surgical scenes. Scale bar equals 100 pixels. 120

Tables

4.1 Attribute and percentage of frames which are tagged with the attribute for each dataset. 69

4.2 Numerical results for the 3D tracking for each of the *ex vivo* sequences. Each value shows the mean error (mm) of the translation error for our 2D3D method and for the 3D only tracking. 74

5.1 The performance of the proposed algorithm compared with different low-level features. PAWSSa and PAWSSb tracker represents our tracker without and with the KLT tracker, respectively. 84

5.2 The performance of the proposed algorithm and the SOWP tracker [2] compared with the Struck tracker [5] 84

5.3 Comparison of the PR/SR score in the OPE based on the 11 sequence attributes: illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background cluttered (BC) and low resolution (LR). The best results are shown in **bold**. 85

5.4 Comparison of the PR/SR score based on the 11 sequence attributes with state-of-the-art trackers in the OPE. For the descriptions of the challenging factors, refer to the caption of Table 5.3. The best and the second-best results are shown in **red** and **blue** colours respectively. 85

5.5 VOT2014 results. The best and the second-best results are shown in **red** and **blue** colours respectively. 89

5.6 VOT2014 without re-initialization results. The best and the second-best results are shown in **red** and **blue** colours respectively. 89

5.7 VOT2015 score/ranking and expected overlap results from the top trackers of VOT2014, VOT2015 and the baseline tracker. The NCC tracker is the VOT2015 baseline tracker. Trackers marked with † are submitted to VOT2015 without publication. 92

5.8 Accuracy of *EndoVis* Articulated Robotic Surgical Instrument Train Data for the TrackedPoint 95

5.9 Accuracy of *EndoVis* Articulated Robotic Surgical Instrument Train Data for HeadPoint and ShaftPoint joints with High Quality Annotation 96

5.10 Distance (pixel) comparison with all the submitted methods for the TrackedPoint of the robotic laparoscopic instrument test set. Multiple instrument challenging subset is evaluated separately. 97
5.11 Distance (pixel) comparison with all the submitted methods for the TrackedPoint of the conventional laparoscopic instrument test set. Various challenging subsets are evaluated separately. 97

6.1 The Network Specifications for the Detection Subnetwork: The Kernel Size and Stride, and the Output Size (Channel $\times$ Height $\times$ Width) of Each Layer. The Original Dimension of the Input Image is $3 \times h \times w$, and the Network Outputs stacked $(M + N)$ Probability Maps with the Same Size as the Input Image. . . 106
6.2 The Network Specifications for Regression Subnetwork: The Kernel Size and Stride, and the Output Size (Channel $\times$ Height $\times$ Width) of Each Layer. The Regression Network is Fed with the Concatenation of the Input Image and the Detection Output Maps, and Outputs stacked $(M + N)$ Probability Maps with the Same Size as the Input Image. 108
6.3 Label / Frame Number of the *EndoVis* and *RMIT* Dataset 111
6.4 Quantitative Results of the RMIT Dataset: Precision and the Distance Error Between Ground Truth and the Estimate of Each Joint. The Threshold is Set to 15 Pixels for the Original Resolution of 640×480 Pixels. 113
6.5 Quantitative Results of the RMIT Dataset: the Strict PCP Score of the Estimate of Each Joint Pair. 113
6.6 Quantitative Recall Performance Comparison with the State-of-the-art Methods on the RMIT Test Set[1] 114
6.7 Quantitative Strict PCP Score Comparison with the State-of-the-art Methods on the RMIT Test Set 114
6.8 Quantitative results of the *EndoVis* dataset: precision and the distance error between GT and the estimate of each joint. For the *EndoVis* dataset, the thresholds are set to 20 and 30 px for the original and smoke-simulated test data with the resolution of 720×576 px. 115
6.9 Ablation Study of the Detection-Regression Model Architecture on EndoVis Test Set 118
6.10 Quantitative results of the *in vivo* dataset: precision and the distance error between GT and the estimate of each joint. For the *in vivo* data, the threshold is set to 50 px for the original resolution of 1920×1080 px. 119

Acronyms

AR	Augmented Reality
AUC	Area Under Curve
CAS	Computer Assisted Surgery
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CT	Computed Tomography
DLK	Deformable Lucas-Kanade
DOF	Degree of Freedom
dVRK	da Vinci Research Kit
FCN	Fully Convolutional Network
GHT	Generalized Hough Transform
GT	Ground Truth
HOG	Histogram of Oriented Gradients
KLT	Kanada-Lucas-Tomasi
LK	Lucas-Kanade
MIS	Minimally Invasive Surgery
MRI	Magnetic Resonance Imaging
MSI	Multispectral Imaging
NMS	Non-maximum Suppression
OPE	One Pass Evaluation
OR	Operating Room
OTB	Online Tracking Benchmark
PCP	Percentage of Correct Parts
PFN	Progressively Finite Newton
PR	Precision Rate
RANSAC	Random Sample Consensus
RMIS	Robotic Assisted Minimally Invasive Surgery
ROI	Region of Interest
SCV	Sum of Conditional Variance
SIFT	Scale Invariant Feature Transform
SR	Success Rate
SSD	Sum of Square Difference
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
TPS	Thin-Plate Spline
US	Ultrasound
VOT	Visual Object Tracking
VR	Virtual Reality

Chapter 1

Computer Assisted and Robotic Minimally Invasive Surgery

1.1 Minimally Invasive Surgery

The field of surgery has always evolved by exploring new techniques to make the surgical procedures safer and more effective. In minimally invasive surgery (MIS) procedures, the surgical site is visualized by using endoscopic and laparoscopic white light cameras. Historically, the field was inspired once Harold Hopkins introduced the solid glass rod lens which hugely improved the image transmission from the surgical site. Karl Storz patented the rod lens and coupled it with cold light fibre optics for illumination. The joint Harold and Storz rod-lens scope was introduced into MIS in the 1960s, causing a breakthrough and paving the path towards the advanced endoscopes used nowadays [7]. Several pioneering surgeons progressively performed clinical assessments of MIS in the 1970s and 1980s, Professor Erich Mühe performed the first laparoscopic cholecystectomy in 1985, and eventually positive long-term outcomes from endoscopic surgeries promoted MIS and the establishment of endoscopic centres throughout the world. In a modern operating room (OR), MIS has often replaced the conventional open surgery approach and has become the procedure of choice across many surgical disciplines. Figure 1.1 shows how surgeons perform MIS using a laparoscopic approach using elongated instruments get access to the internal organs, visualising the surgical site displayed on a 2D monitor using video cameras and rod lens optics (or chip-on-tip sensors). The use of small incisions results in less trauma, faster recovery and shorter hospitalisation time and decreased risk of co-morbidity for patients.

Despite the fact that endoscopic procedures are popularized throughout the world, keyhole surgery has its own ergonomic limitations, such as the fulcrum effect and impaired dexterity. Conventional rigid instruments lack wrist articulation at the instrument tip, they are restricted within a small range of motion and have a limited workspace. Inspired by the way advances in fibre optics and medical imaging resulted in MIS, the community kept seeking new ways to alleviate the limitations of MIS especially using robotics and computing. In the late 1990s, robotic assisted minimally invasive surgery (RMIS) systems developed through research initiatives began clinical use [8]. The integration of robotic precision and flexible human control overcomes many of the problems associated with traditional MIS. The fulcrum effect is eliminated through the digital master-slave setup and additional flexibility such as wrist articulation

Figure 1.1: Surgeons performing laparoscopic cholecystectomy in a modern operation room with minimally invasive techniques. The use of laparoscope and elongated instruments passing through the access ports give the surgeons access to the internal organ and visualization of the surgical site on a 2monitor.

can be introduced to the instruments.

The most successful robotic surgery platform to this day is the da Vinci® (Intuitive Surgical Inc., CA), which is discussed in more detail in the next section. The surgical system assists the surgeon by translating the surgeon's hand movements into precise movements of dexterous instruments inside the patient's body. Using the da Vinci, the integration of robotics and MIS has made impressive progress. MIS continues to be driven towards further minimization of the number and size of incision are both reducing. Laboratory and clinical application of natural orifice transluminal endoscopic surgery [9] and laparoendoscopic single-incision laparoscopic surgery [10] has been introduced. Until now, the widely used robotic platforms utilize long, rigid instrument with fixed placement and restricted effective workspace, which limits the surgical procedures involving complex anatomical pathways. This motivates the next era of MIS flexible access surgery [11]. The requirement of flexible access surgery is to get access to different target anatomy from sites that are not aligned in the most convenient or ergonomically optimum positions. Thus, one of the main research focuses of recent medical robotics is on the development of articulated and flexible robots for complex transluminal and single port techniques, with the associated ergonomic and safety requirements during the robotic assisted surgery.

1.2 Computer Assisted Interventions in Minimally Invasive Surgery

For a surgeon, MIS has brought new challenges and novel surgical technologies have changed surgical routines. In the meantime, requirements for surgeons to keep up-to-date and familiarize with the latest platforms need to balance with clinical practice [12]. Mastering complex surgical instrument control from the surgeon's point of view is challenging due to the restricted view and the lack of tactile feedback. During surgery, the two most important senses surgeons rely upon are sight and touch [13]. To master the skill of operating rigid or flexible but unstable instruments within a limited operative workspace that is viewed on an independent monitor requires a high level of hand-eye motor skill, dexterity and coordination, which results in a steep learning curve.

The introduction of robotics and articulated instruments into MIS has brought additional benefits to alleviate this learning curve, such as enhanced dexterity, reducing the manual precision down to micro scale, but also associated safety and effectiveness concerns. In the meantime, vision as the main feedback from the surgical site is the most important clue for surgeons to operate with. Therefore, platforms for incorporating image guidance and effective imaging and vision methods are critical development areas needed to support surgeons in having all the important information for navigation [14].

Image-guided surgery uses the main idea that pre-operative information, usually from Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), can be used to identify anatomical landmarks and these can be registered to the operative view during a procedure [15]. For example, surgeons make surgical plans on pre-operative data, then during surgery, patient specific models provide additional structural or pathological information about the patient registered with intra-operative imaging and the surgeon performs the surgery guided by the pre-procedural planning. Although the idea behind this form of Image-guided surgery has been studied for a long time, the problem of intra-operative deformable registration of multi-modal information is still a major challenge.

1.2.1 Intra-operative Imaging Techniques

For intra-operative imaging, limited modalities are available to deliver real-time information to the surgeon. Imaging techniques such as multispectral imaging (MSI) or interventional MRI can be introduced in surgical procedures to provide additional characteristic, functional information about the surgical site but they are either difficult and unreliable to implement or very expensive and require special non-ferromagnetic instrumentation. Though CT has excellent solid organ contrast (i.e., bones) and high spatial resolution, it is not realistic to expose patients and surgeons high dose of radiation that lasts for a long intervention. Real-time interventional MRI has been introduced for guidance for neurosurgery [16, 17] and cardiac procedures [18, 19] thanks to its non-invasive imaging modality and its versatile tissue contrast. However, there is a trade-off between the spatial and temporal resolution, the increased acquisition frame-rate is obtained at the expense of imaging quality and spatial coverage. On the other hand, high spatial resolution images would sacrifice real-time performance. Without the organ dynamics being fully captured, it significantly reduces the targeting accuracy of interventional procedures. Besides,

limited devices are available to be used safely in an MR environment.

Ultrasound (US) is relatively low-cost and is suitable for intra-operative use. Intra-operative 2D US has been widely used in MIS and RMIS to visualize beneath the exposed tissue surface. Compared to other modalities, such as intra-operative MRI, US is cost-effective and portable, and it offers real-time structural (or functional if Doppler) imaging, without adding much surgical complexity or time. In Figure 1.2 (a), da Vinci robotic system is guided by US to localize a tumour and determine the resection margin during partial nephrectomy[1]. However, conventional 2D US, is highly dependent on the experience and knowledge of the surgeon. 3D US was introduced to overcome the limitations by providing 3D volume reconstruction which help the surgeon establish a full understanding of spatial anatomic structure. Therefore, many efforts have been made to develop real-time or near real-time US systems in recent years. However, no matter 2D or 3D, the low image quality of US due to speckle, poor signal-to-noise ratio and various imaging artefacts has hindered the wider application of this technique, for example, surgeons find it challenging to navigate instruments in the dynamic, constrained space, especially with 3D US, hard objects such as instruments looks distorted. In neurosurgery, intra-cardiac procedures or hepatic ablative therapy, US images is usually registered with pre-operative high resolution imaging such as CT or MRI to improve the navigation accuracy. Auxiliary robotic arm has been introduced to hold the US probe to increase the accuracy and reducing the surgeon's cognitive workload. Recently, Mathiassen et al. [20] developed a tele-operated robotic US system using a low cost commercial robot shown in Figure 1.2 (b).

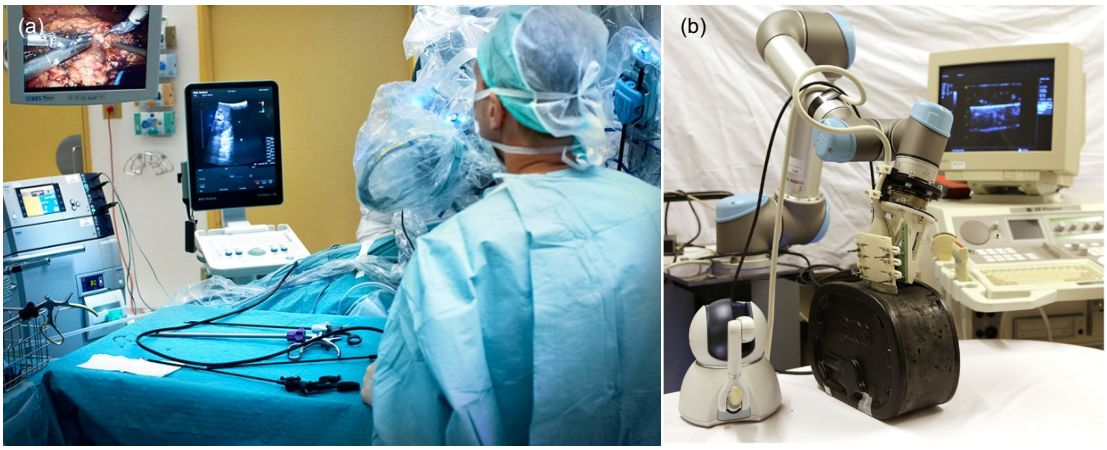


Figure 1.2: (a) An example illustration of the da Vinci surgical system guided by US during robotic-assisted partial nephrectomy; (b) Robotic US system using commercial robot [20]: in the centre the robot is holding a US probe during experiments on an abdominal phantom.

MSI, as an new optical modality, includes the acquisition of a stack of 2D images of reflected light sampled at different wavelengths. In retinal surgery, it could be used for reducing the phototoxicity exposure [21]. In endoscopic procedures MSI and the data cube of spectral information can be used together with the complete spectrum of the tissue in order to estimate the concentration of chromophores in a tissue area. Through analysing the data cube, critical information about the hemoglobin and tissue oxygenation distribution can be inferred, which

[1] `http://blog.bkultrasound.com/the-benefits-of-ultrasound-in-robotic-surgery`

could be valuable for applications such as diagnosis of mesenteric ischaemia, and treatment assessment of the bronchial tumours [22, 23].

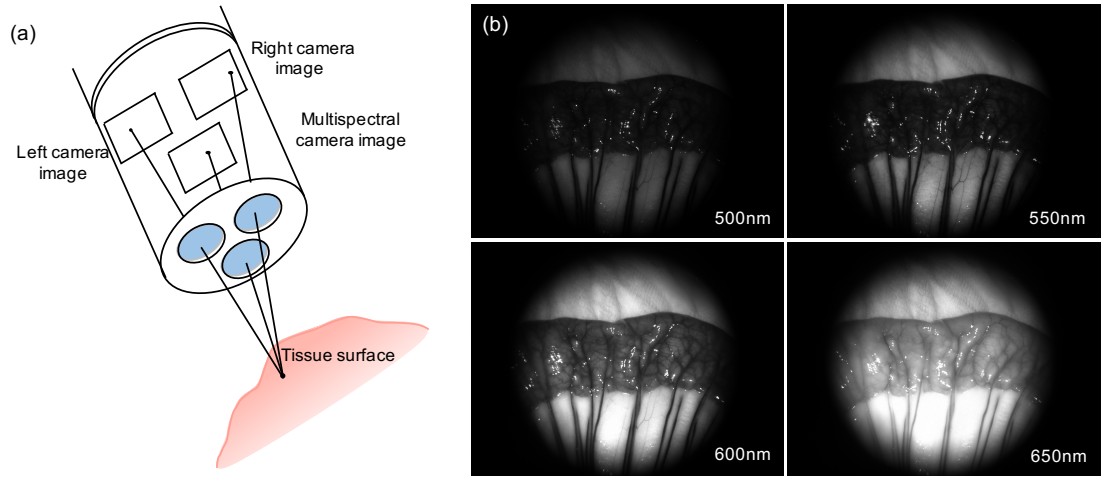


Figure 1.3: MSI techniques: (a) Trinocular endoscope system [24]; (b) Example single band spectral images of the small bowel serosa acquired at different wavelengths within 500-650 nm.

1.2.2 Towards Soft Tissue Surgical Navigation

Back in 1986, to overcome the limitations of frame-based stereotaxy, David Roberts introduced the concept of frameless stereotaxy for neurosurgery [25, 26]. In the system, an operating microscope was tracked in 3D by a sonic digitizer, and the target location on pre-operative MR or CT images is projected to the microscope ocular. It shaped the conception of navigation in surgery. In a typical neurosurgery today, reflective marker spheres are attached to both the patient and the instruments, the setup enables the pre-operative data and the instrument to be registered rigidly in the patient's coordinates, during surgery the instrument can be tracked in real time and visualized on the pre-operative images. Not only it relieves the patient from the significant discomfort of fixed head frames, it provides valuable intra-operative guidance for the surgeon during the procedure to locate the instrument and the anatomical target, and to avoid critical structure. We show the advanced commercial optical track system and intra-operative MR and US imaging from Brainlab AG[2] in Figure 1.4.

With the development of medical imaging and robotics, navigation has emerged in Ear, Nose and Throat surgery, knee or hip replacements and spine surgery. In recent years, navigation is not restricted to almost rigid organs, but is extended to soft tissue in endoscopic thoracic and abdominal surgery [27, 28]. The commercial da Vinci surgical system has been used worldwide for various minimally invasive interventions such as cardiac, thoracic and general surgeries. Intuitive Surgical Inc. also released the da Vinci Research Interface which allows researchers or third-party developers to retrieve stream of kinematic and user event data from the robot [29]. Since they may provide stable optic, standard user interface and instrumentation, positional information retrieved from kinematics, the increasing involvement of robots and telemanipulators in surgical procedures has influenced and benefited the development and progression of surgical navigation systems. More and more approaches have been proposed

[2] `https://www.brainlab.com`

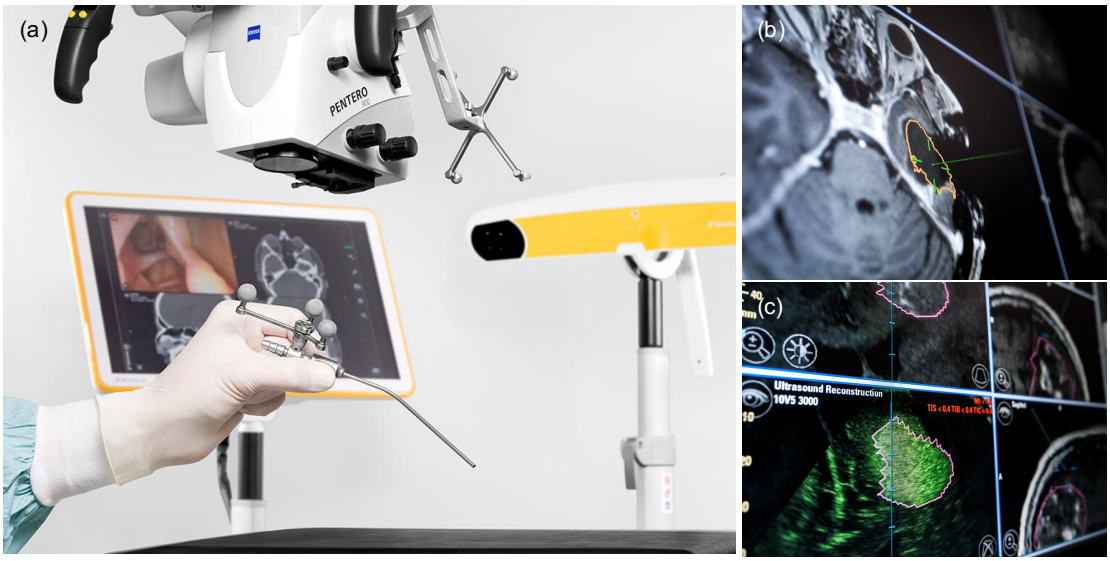


Figure 1.4: Commercial optical tracking system from Brainlab AG (a) with softwares providing intra-operative MR imaging integrated with surgical planning (b) and intra-operative US overlaid with pre-operative data (c).

for navigation with surgical robots. Falk et al. presented video overlay to enhance the intra-operative orientation in cardiac surgery by utilizing the positional data of the da Vinci system. Leven et al. [30] presented a telerobotic surgical system which integrated a laparoscopic ultrasound probe with the da Vinci robot, the US image is displayed in real-time through the surgeon console, enhancing surgeon situational awareness and assisting needle targeting procedures. Buchs et al. coupled the da Vinci Si System to the 3D navigation system CAS-One (CAScination AG, Bern, Switzerland) [31] to provide a real-time augmented endoscopic view within the da Vinci console. The models of both tumour and the instrument were displayed during the liver resection to minimize the risk of positive resection margin [32]. The da Vinci Research Interface The research of these navigation methods in return benefits the practical use of robotic systems in the long term.

The da Vinci system shown in Figure 1.5 includes several key components: a surgeon console, a patient-side cart with interactive robotic arms, a high-definition (HD) vision cart, and specially designed *EndoWrist*® instruments (Figure 1.5 (d1-4)). It is powered by advanced robotic technologies and allows the surgeon's hand, wrist movements to be scaled, filtered and transformed into movements of the instruments with multiple degrees of freedom (DOF) working inside the body of the patient shown in Figure 1.5 (b). The patient lies where the patient-side cart during surgery. The cart includes either three or four robotic arms that carry out the surgeon's commands. A surgeon views 3D image inside the patient's body through the display of the console, and the console translates the surgeon's hand, wrist movements into real-time, precise movements of the surgical instruments (Figure 1.5 (c)). Besides, a vision system is equipped with a 3D endoscope and image processing system that provides images of the patient's anatomy. The vision cart provides a broad perspective and visualization of

[3] `http://internationalbusinessfestival.com/news/shafi-ahmed-vr-ar-in-surgery-medical-realities`

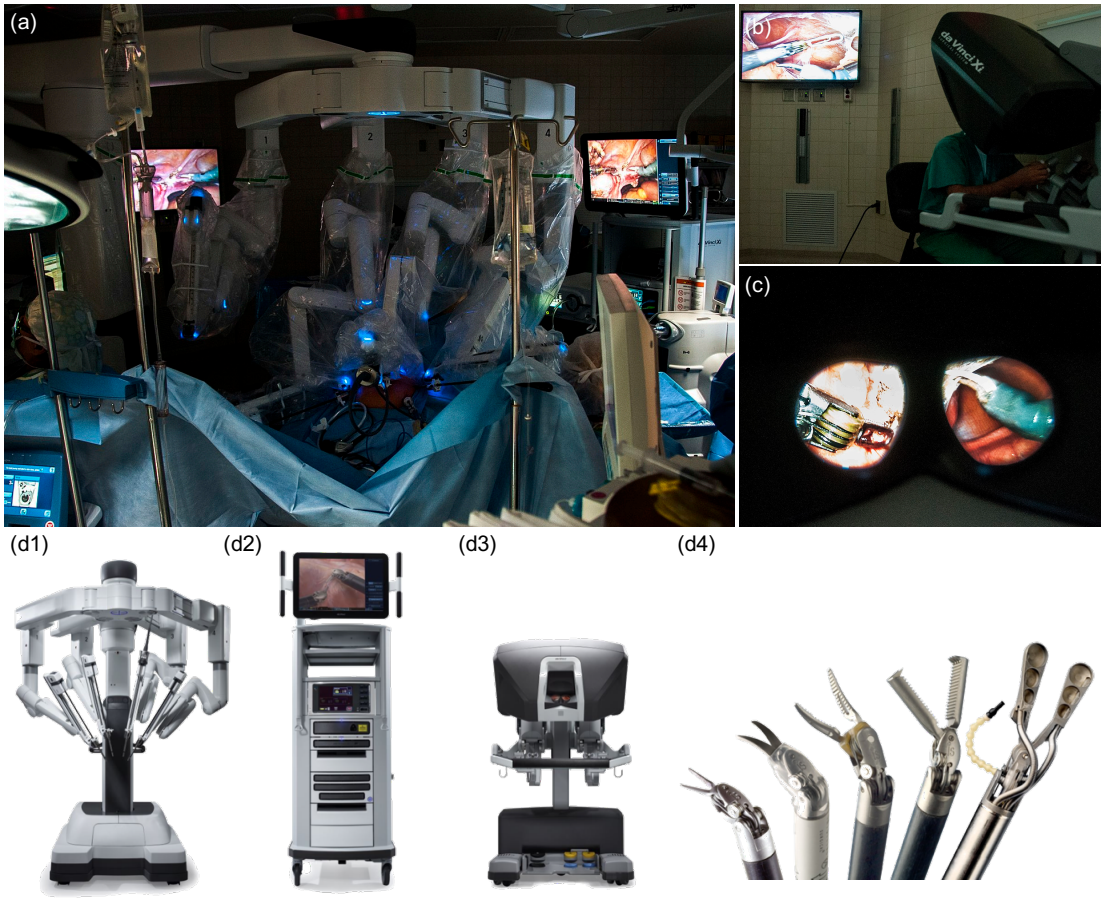


Figure 1.5: (a) Robotic surgery performed using the state-of-the-art da Vinci Surgical® System (Intuitive Surgical Inc., CA); (b) The surgeons looks through via the console mirroring the monitor while operating with articulated instruments on the patient; (c) Surgeon's view through the imaging system displaying articulated instruments. The system includes (d1) Patient-side cart; (d2) HD vision cart; (d3) Surgeon console; (d4) *EndoWrist*® instruments; Surgeons perform surgery by manipulating the robotic system[3].

the surgical procedure to the entire OR team (Figure 1.5 (b)). Although the system is quite costly, the greatest strength of the system is that it restores the wrist articulation lost during conventional endoscopy, which is considered to have provide increased precision and enhanced dexterity [8, 33]. A full range of instruments are designed for specific tasks in da Vinci system, such as clamping, cutting, manipulating tissue shown in Figure 1.5 (d4). Besides the high cost and large space requirement for the robot, one of the disadvantages of the system is that the port-placement and effective workspace is restricted due to the rigid shaft of the surgical instrument. Regarding the problem, novel instruments such as *VeSPA* have been introduced for the modified da Vinci system to perform single-site surgery [34]. Compared to *EndoWrist* instruments, the shaft of *VeSPA* is semi-rigid, allowing them to be inserted via curved cannula, although problems such as instrument clashing, lack of robust retraction and surgeon ergonomic discomfort remain to be technically challenging.

Besides surgical navigation systems, many efforts have been devoted to improving surgical skills training. Trainees benefit a lot from simulation considering the surgical risks and the range of various techniques which they are expected to master. Different technologies such as 3D printing, virtual reality (VR) and augmented reality (AR) have brought new possibilities

into the training schemes. The development of medical imaging has promoted the medical application of 3D printing. For example, we can import CT images into 3D printing software and create anatomically accurate models for different organs. Besides physical models, trainees can also perform procedures on virtual organs in a computer generated environment. Complicated training software platforms are designed using VR technique to offer realistic interactions in an immersive environment. Mirracle[4], an AR magic mirror system developed in 2011, allows the user to virtually look inside their own body when standing in front of the system [35]. As shown in Figure 1.6, it uses Microsoft's Kinect to track the user's pose and augment the medical data such as 3D model of anatomy, onto the user's body. Currently, the system has been transplanted on the Balaur Display wall, which is a high resolution display system intended for research in multi-user gestural interaction with large imagery and datasets. By using the Balaur Display wall as a display device, this opens up new opportunities for Mirracle for various applications such as medical and healthcare education and AR rehabilitation.

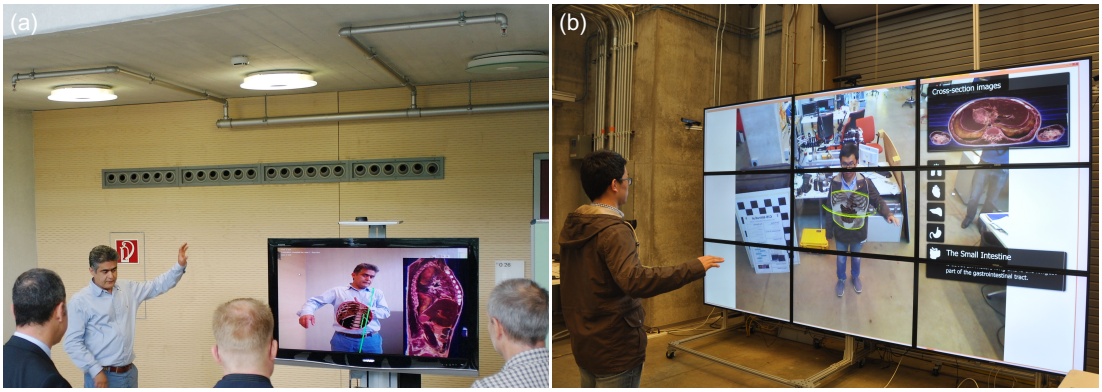


Figure 1.6: The AR Magic Mirror system: (a) the Mirracle system prototype; (b) Mirracle transplanted on the Balaur display wall.

1.2.3 Effective Image Guidance

Either registration of pre-operative and intra-operative data or reconstruction of intra-operative data has to face the challenges in the modelling of dynamic, deformable soft tissues, especially for the abdominal cavity undergoing laparoscopic surgery. The surgical instruments interacting with the organ may cause large occlusion and deformation, illumination changes due to the laparoscopic light, smoke or bleeding during surgery may affect surface deformation recovery. Besides, the most difficult associate issue is the real time modelling of organ deformation.

Despite all the advantages of MIS, surgeons are sometimes challenged by the restricted surgical site. Narrow space and blood obscuring the visibility are factors limiting the surgeon's ability to locate the exact position of the instruments. To overcome the limitations of hand-held instruments, robotics is introduced to MIS, providing dexterous and tremor stabilizing tools. Of course, the robotic system allows finer, tremor-free motion control of the instrument, in the end, it depends on the manipulation skill of the surgeon. Effective intra-operative instrument localisation has demonstrated its clinical potential for enhancing the functionality of MIS and also by avoiding pre-defined safety regions, it can facilitate safe operation. Besides, it also

[4] http://campar.in.tum.de/files/mirracle/

applies to the training and assessment of surgical skills. For example, part of the surgical skill training and assessment depends on the analysis of efficiency metrics that are based on the instrument kinetics and dynamics during performing the task. Computer vision based tracking systems can generate these metrics by analysing the endoscopic surgical videos. Their non-invasive and unobtrusive nature make them suitable for training purpose and offline analysis.

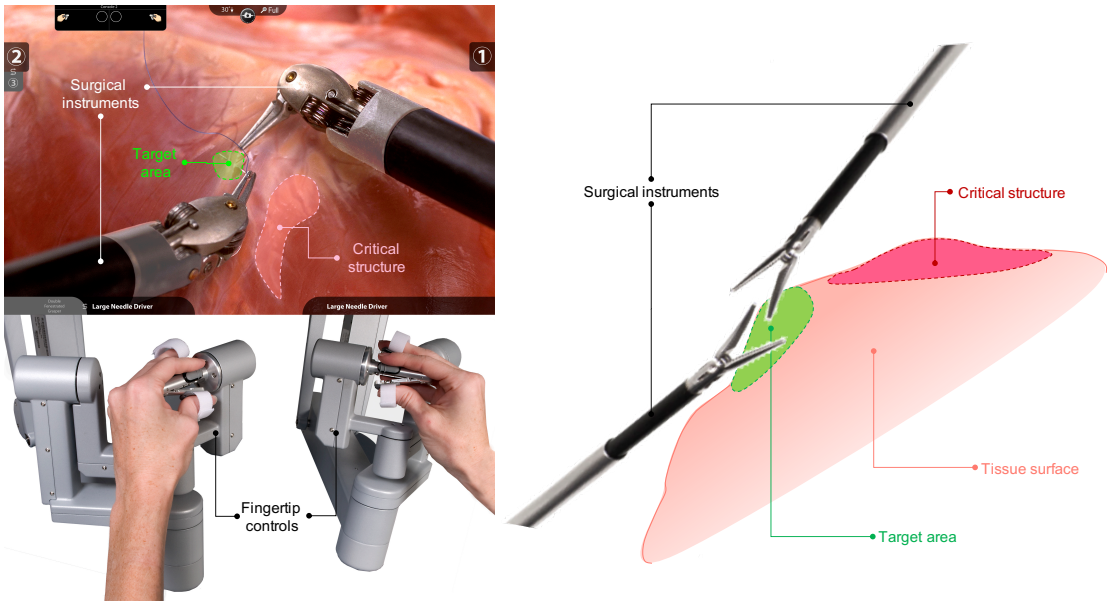


Figure 1.7: Vision techniques integrated with image guidance illustration: (Left) displays 2D guidance from the view of the surgeon with the location of the instrument, the pre-defined target area, and critical structure to be avoid; (Right) corresponding 3D surgical guidance with tissue surface model.

1.3 Thesis Structure

The thesis is organized into chapters as follows:

Chapter 2 provides a review of the main theory, methodologies and technologies relating to general visual tracking frameworks. In the spirit of conciseness, a general visual tracking framework is decomposed into block such as visual representation, observation model, motion model and model adaptor based on their individual role and function. For each component, a review of its goal, development and contributions to the entire system, is provided. A focused review of applied research on visual tracking in endoscopic surgery is then provided, particularly deformable surface estimation, instrument tracking and pose estimation, presenting the technical challenges and the state-of-the-art.

Chapter 3 investigates the application of recovering surface deformation during robotic-assisted MIS procedures. Conventional methods have their limitations when dealing with illumination changes, poor textural information or complex reflectance, so we propose our hybrid tissue surface deformation estimation method, which combines the advantages of both sparse feature and dense intensity information to track the tissue robustly and reliably. The algorithm is thoroughly validated on both synthetic data with known ground truth (GT) and on *ex vivo* and *in vivo* endoscopic dataset recorded from da Vinci platform, and on multispectral image sequences as well.

Chapter 4 discusses surgical instrument tracking problems in computer-assisted interventions for MIS. Vision-based approaches are promising with minimal hardware integration requirements, but in the meantime, these methods may result in drift or tracking failure under occlusion, shadows and fast motion. We develop a 2D Generalized Hough Transform (GHT) based tracker using keypoint features, the tracker can both handle complex environmental changes and recover from tracking failure. The tracker is used as the initializer to a pre-existing 3D tracker for full pose estimation of the surgical instrument over long sequences. The whole 2D-3D tracking framework achieves drift-free, robust and accurate performance on both *ex vivo* and *in vivo* data, suggesting that combining 2D and 3D tracking is a promising solution to deal with complex situations in surgical instrument tracking.

Chapter 5 continues the research on instrument tracking of last chapter but from another point of view. Based on tracking-by-detection algorithms, the tracking problem is treated as a classification task, and the object model is updated over time using online learning techniques. These methods are prone to include background information in the object appearance or lack the ability to estimate the scale changes which degrades the performance of the classifier. We incorporated patch-based visual representation and a colour-based segmentation model to adaptively suppress the background information. We validate it on *in vivo* surgical instrument sequences. The framework also can be applied for any general 2D tracking task.

Chapter 6 presents research on pose estimation of articulated surgical instruments. Deep learning technologies have proved their success for different visual tasks in the last few years. Two of the important factors are the availability of large-scale annotated datasets and deep convolutional neural networks. Applying deep neural networks to endoscopy has been challenging due to the lack of related large-scale annotated dataset. Therefore, we propose a dataset with high-quality joint annotations for pose estimation tasks in endoscopy. And we develop a deep neural network for articulated multi-instrument 2D pose estimation trained on our proposed datasets. Our framework is tested on both *ex vivo* and *in vivo* data with promising results. The annotation and the results provide a solid baseline for future work for this task.

Chapter 7 summarizes the thesis, discusses the limitations and also outlines possible directions for future research.

1.4 Contributions

The purpose of this thesis is to provide technical solutions for visual tracking in image-guided surgery. Computer vision based technologies are incorporated for the image guidance to extract information about the surgical site: non-rigid tissue surface deformation, the location and the pose of the surgical instrument are estimated in the framework.

The work presented in this thesis has resulted in the following publications:

- X. Du, N. Clancy, S. Arya, G. B. Hanna, J. D. Kelly, D. S. Elson, and D. Stoyanov, Robust Surface Tracking Combining Features, Intensity and Illumination Compensation, *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 12, pp. 1915-1926, 2015. [Chapter 3].

- N. T. Clancy, S. Arya, D. Stoyanov, X. Du, G. B. Hanna, and D. S. Elson, Imaging

the Spectral Reflectance Properties of Bipolar Radiofrequency-fused Bowel Tissue in *European Conference on Biomedical Optics*, p. 953717, Optical Society of America, 2015.

- S. Arya, N. T. Clancy, D. S. Elson, G. B. Hanna, D. Stoyanov, and X. Du, Surgical Device with an End Effector Assembly and System for Monitoring of Tissue Before and After a Surgical Procedure, May 25 2016. US Patent App. 15/164,701.

- X. Du, M. Allan, A. Dore, S. Ourselin, D. Hawkes, J. D. Kelly, and D. Stoyanov, Combined 2D and 3D Tracking of Surgical Instruments for Minimally Invasive and Robotic-assisted Surgery, *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 6, pp. 1109-1119, 2016. [Chapter 4].

- X. Du, A. Dore, and D. Stoyanov, "Patch-based Adaptive Weighting with Segmentation and Scale (PAWSS) for visual tracking", *arXiv preprint arXiv:1708.01179*, 2017. (Submitted to *Medical Image Analysis*). [Chapter 5]

- T. Kurmann, P. M. Neila, X. Du, P. Fua, D. Stoyanov, S. Wolf, and R. Sznitman, Simultaneous Recognition and Pose Estimation of Instruments in Minimally Invasive Surgery in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 505-513, 2017.

- G. Jones, N. T. Clancy, X. Du, M. Robu, S. Arridge, D. S. Elson, and D. Stoyanov, "Fast Estimation of Haemoglobin Concentration in Tissue Via Wavelet Decomposition" in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 100-108, 2017.

- X. Du, T. Kurmann, P-L. Chang, M. Allan, S. Ourselin, R. Sznitman, J. D. Kelly, and D. Stoyanov, "Articulated Multi-Instrument 2D Pose Estimation Using Fully Convolutional Networks" *IEEE Transactions on Medical Imaging*, 2018. (Accepted). [Chapter 6].

- V. Penza, X. Du, D. Stoyanov, A. Forgione, L. S. Mattos, and E. D. Momi, "Long Term Safety Area Tracking (LT-SAT) with Online Failure Detection and Recovery for Robotic Minimally Invasive Surgery" in *Medical Image Analysis*, vol. 45, pp.13-23, 2018.

- C. D'Ettorre, G. Dwyer, X. Du, F. Chadebecq, F. Vasconcelos, E. D. Momi, D. Stoyanov, "Automated Pick-up of Suturing Needles for Robotic Surgical Assistance" in *IEEE International Conference on Robotics and Automation*, 2018. (Accepted).

Chapter 2

Visual Tracking and Applications in Computer Assisted Surgery

2.1 Visual Tracking Framework

Visual tracking is one of the most significant computer vision research area with a wide range of applications such as human-computer interaction, video surveillance and endoscopic medical imaging, to name a few. Typically, the tracking target is manually selected or automatically detected with bounding box, ellipse or polygon in an initial frame, and the goal of tracking is to estimate the state (e.g., position, size, or contour) of the target in subsequent frames. Despite extensive research, ranging from the early, simple but effective Lucas-Kanade (LK) method [36, 37], to recent deep learning based trackers [38, 39, 40], visual tracking remains a challenging problem. Multiple factors may contribute to the performance degradation of a tracking system, including (i) sensor quality (e.g. low resolution or frame rate, colour distortion); (ii) general appearance variations (e.g. illumination changes, motion blur, partial or full occlusion, object deformation, scale variations); (iii) application specific requirements(e.g. non-rigid tracking, irregular object contour definitions, multiple objects, cluttered backgrounds); (iv) processing speed requirements. Such factors test the robustness of a tracking algorithm and need to be considered in the design of a functional tracking system. In surgery, challenges arise due to the wet, dynamic environment and the limited scope for controlling the camera size and position within the body of a patient.

A typical tracking algorithm framework involves components for visual presentation, an observation model, a motion model and a model adaptation strategy [41, 42, 43]. The following sections outline the main lines of thinking behind such algorithm blocks and the current state of the art.

Visual Feature Representation usually focuses on how to design a robust representation for the tracking target's appearance in images using different image features. A good representation should be expressive, which means it should capture salient information that differentiates a tracking target from the background scene, and generally be insensitive to noise or local variation like articulation and global changes like illumination.

Observation Models are used for target identification through extracted feature representations. Building effective mathematical models using statistical learning techniques is a common

approach, which can be classified into generative, discriminative or hybrid categories. Generative models focus on learning a compact target model, then search for the most similar candidate in image space. On the other hand, discriminative models pose target tracking as a binary classification problem by maximizing the difference between the target and background.

Motion Models are used to generate estimates of possible locations of the tracking target in time by making assumptions on the continuity of motion or utilizing various search schemes, such as sliding window, Kalman filters [44] and particle filters [45, 46].

Model Adaptors are instrumental for robust tracking applications because the appearance of the tracking target commonly changes over time , either gradually or abruptly. The model adaptor controls when and how the observation model updates. The adaptation strategy should keep balance between computational speed and drift failure.

2.1.1 Visual Feature Representation

Methods for visual object tracking are highly dependent on the choice of feature representation. Extensive research has been devoted to data pre-processing pipelines and feature extractors which results in a effective data representation with favourable properties such as invariance and uniqueness. More recently, machine learning strategies have become a standard for avoiding hand crafted representations and achieving robust performance. Broadly, speaking visual representation approaches can be categorized into global and local features.

A global feature representation reflects the global statistical properties of the target's appearance. Raw image data, pixels, are usually transformed into more compact and informative representation. Local feature representations usually utilize distinctive local information such as keypoints or image patches to encode the target appearance. A local feature is an image pattern which is different from its neighbourhood. Experiments have shown that local features are important visual clues in our biological vision system. For example, removing edges or corners of the object in a image affect more than removing smooth lines for human object recognition [47]. The pattern could be associated with various image properties, such as colour, texture or shape. Local features could represent specific interpretation, such as edge or blob detectors. But they are not necessarily always related to target representation. A set of anchor points could be used for object recognition as long as they could be localized and identified robustly over time. Typical feature representations can be categorized into these categories:

Raw Image Pixels are a simple but effective representation used extensively in early vision. The target ROI is resized to a fixed size, and the feature is represented by the flattened vector of pixel values in a particular colour space or in greyscale. Except for the widely used vector-based representation form, 2D or higher order tensors are also constructed as the matrix-based feature description, which is relatively low-dimensional.

Gradients represent the directional change in the intensity of pixels within the image and form the classical cue for detecting saliency. An early example is the Canny edge detector [48], which takes advantage of the gradient information to detect edges as areas with strong intensity contrasts. Because gradients can often align with object boundaries, they are often useful

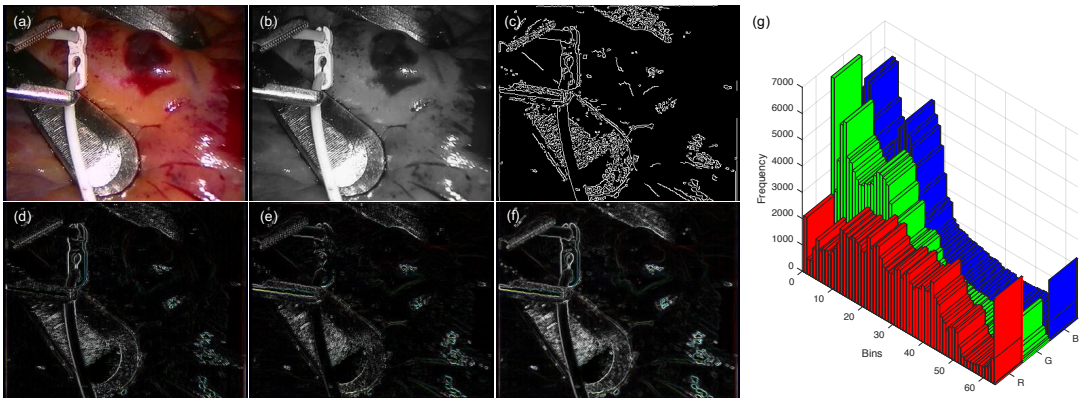


Figure 2.1: Examples of image feature representation: (a) raw image pixels in colour space (b) or greyscale; (c) Canny edge detection result; (d) absolute value of x-gradient; (e) absolute value of y-gradient; (f) magnitude of gradient; (g) colour histogram of red, green and blue channels separately.

to define their extent and an example of this approach is the active contour representation [49] commonly used in non-rigid tracking by delineating the object outline. Compared to the bounding box representation, active contours have the ability to represent a more complicated shape, which segments the object from background. The deformable contour is evolved by internal forces that tend to penalize deformation and image forces which pull towards object boundary.

Histogram representations capture the distribution of pixels inside the target image region. Rather than raw image pixels, the distribution approach potentially encodes higher level information within the region. Histogram binning can be constructed in different colour spaces, such as RGB, HSV or Lab, or in greyscale. On one hand, colour histograms do not consider the structural information of the target, which makes them robust for deformable objects, on the other hand, this can also lead to tracking failure by drifting towards similar colour distractors in the background. Histogram of oriented gradients (HOG) is one of the commonly used shape descriptors which describes the distribution of intensity gradients or edge directions [50]. Compared to other descriptors, it has better invariance to illumination changes and shadowing, geometric and photometric transformations. Naturally, multi-cue features are introduced to enhance the representation by combining complementary features. Spatial information can be integrated with colour histogram by either jointly modelling with colour or patch-wise concatenation. Joint spatial-colour modelling describes the distribution properties of object appearance in a joint spatial-colour space (e.g., (x, y, R, G, B). While the patch-wise colour feature capture the local information, which will be discussed below in detail. Besides, gradient or edge information can also be incorporated for robust object tracking in a similar way. In [51], the target is represented by kernel-regularized colour histogram and is searched locally in a mean-shift procedure to perform gradient-based optimization.

Optical Flow is the movement of brightness patterns in an image, which is usually represented by dense pixel-wise field of displacement vectors (see tissue example from heart bypass surgery in Figure 2.2). Flow fields arise from the relative motion of the target and its surroundings [52]. Instead of focusing on the static visual characteristics of the object, it assumes local image translational motion and incorporates complementary joint temporal-spatial motion informa-

tion of the target. For the LK method [37], it assumes constant flow in a local neighbourhood, and then the tracking problem is formulated into an optimization framework with regards to the motion parameters, iterative gradient descent methods are used to estimate the motion parameters. In [53], optical flow is used to provides constraints within a deformable model to estimate the shape and motion of face.

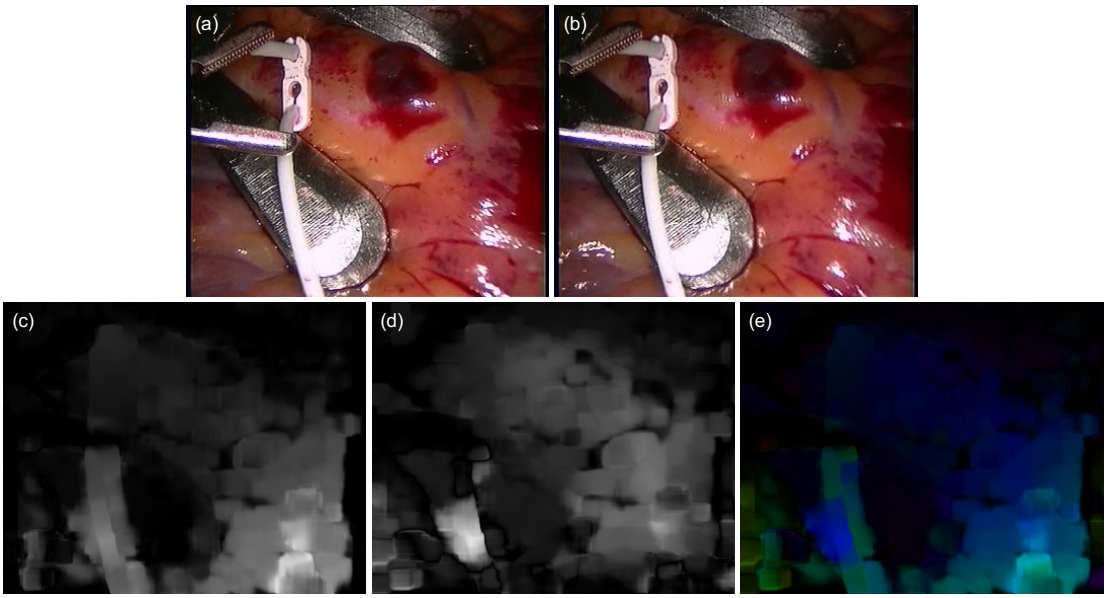


Figure 2.2: Tissue dense optical flow. (a) Frame from heart bypass surgery; (b) The consecutive frame after (c-d) Dense optical flow for x-axis and y-axis; (e) Dense optical flow encoded with colour, in which direction corresponds to Hue value of the image and magnitude corresponds to Value plane.

Template-based Representation represents the target using a set of partition templates. This approach provides a mechanism for coping with partial occlusion or target deformation. The assemble of templates is usually based on spatial relationship or hierarchical decomposition. Patch-wise histogram features can be used to encode spatial information by dividing the target region into a set of patches, and then concatenating the colour/gradient histogram of each patch into the appearance models, such as HOG [50]. The geometric relationship between patches captures the spatial layout information of the target. Another example is part-based models for human detection and pose estimation. The human body is decomposed into a tree structure of body parts. The algorithm detects and locates human by hierarchically matching part templates in a global optimization framework. Besides all these template-based representations which either divide the target into regular grids or rough composition of patches, there are saliency-based methods which select patches based on their discriminative power. The goal of saliency detection is to simulate the human perception mechanism and to detect important regions or objects in the image [54, 55]. By mimicking the selective processing in human visual system, research focuses on why certain regions stand out and get preferable attention from human in a scene, which makes our visual system efficient in tracking moving targets. Among different visual selection strategies, spatial attention extracts local selective blocks with larger discriminative power that any other regions in its spatial as target representation [56].

Keypoint-based Representation uses interest points in the image whose local structure is rich and stable. The development of keypoint detection is closely related to corner detection [57]. Initial works are limited to short range tracking or narrow baseline stereo matching restricted to one scale. Scale-invariant and affine-invariant methods are able to cope with scale changes and general viewpoint changes through using invariant feature detection/description. Usually a feature descriptor is computed for each keypoint to provide distinct information about the region anchored at the keypoint. The descriptor is the core to feature matching for many computer vision tasks such as image understanding, scene analysis or object detection and tracking. It generally uses the combination of local colour or gradient information. Based on the density, it can be divided into two categories: dense or sparse. A dense descriptor takes advantage of all the pixels in the feature region, such as Scale Invariant Feature Transform (SIFT) [58] and Speeded-Up Robust Features (SURF) [59]. SIFT is one of the most widely used keypoint detection and description methods. It computes orientation histograms for multiple regions around the keypoint by computing gradient magnitude and orientation, then the orientation histograms are concatenated into a 128-dimensional vector. While a sparse descriptor only selects subset of the pixels, like binary descriptor methods such as FREAK [60], BRIEF [61] or BRISK [62]. These binary descriptors deploy a pixel-pair comparison strategy, pixel pairs are selected in a specific sampling pattern, and compared to form a binary vector. Since these descriptors encode the feature patch by a binary string, they have fast computation time. Feature detection and description can be chosen separately, but they must work well together for the type of images being processed. Pilet et al. [63] proposed a feature-based tracking framework, in which the target is modelled as a deformable mesh, and the vertices of the mesh are iteratively estimated through a semi-implicit minimization scheme.

Deep Visual Features form the current state of the art in representation with convolutional networks demonstrating best performance at the ImageNet classification benchmark [64]. Features extracted from ImageNet pre-trained convolutional neural networks (CNN) models are used as generic image representations without explicit modelling. Efforts in understanding why large CNNs achieve such impressive performance through visualization techniques have observed that the learned features from different layers in a fully trained model do relate to low level early vision techniques [65]. The first layers of the CNN learn local-level, shallow features, such as corners edges, lines or colour blobs, whereas the latter layers have more complex invariances, capturing class-specific or semantic features. By comparing the changes in feature vectors from the top and the bottom layers of the CNN for images undergoing transformations, it was observed that small transformations have more dramatic effect on the first feature layer than the top feature layer. The discriminative ability of each network layer can be probed by training a linear classifier, like a Support Vector Machine (SVM), on features from different layers and comparisons have shown that higher layers generally generate more discriminative features, which supports the premise that CNNs learn increasingly powerful features as ascending the layers. Additional studies and experiments for different visual recognition tasks and various datasets have investigated the power of CNN off-the-shelf features [66] showing consistently superior performance on multiple tasks compared to the highly tuned state-of-the-

art methods, such as HOG or visual bag of words (BoW), which confirms that CNN feature representation should be considered as the go to approach for visual recognition.

2.1.2 Observation Model

Based on whether to model only the target or both the target and the background, the observation model can be divided into generative and discriminative methods.

Generative trackers learn the target appearance model, and search for the most similar candidate through parameter estimation or optimization. The LK method was one of the important milestones in early generative observation models [36]. It aligns a template image ROI within the search space via minimizing the sum of square differences (SSD) loss. Various extensions have been proposed. Kanade-Lucas-Tomasi (KLT) tracker [67] first detects some good feature points, and used the window-based techniques proposed in [36] to track them in the consecutive frames. The original linear transformation registration is also generalized to more complex warps, such as piece-wise affine transformation [68, 69]. In recent years, with the development of machine learning techniques, more advanced generative model trackers are proposed. One typical generative trackers are kernel based, which construct the target representation by a convolution of the features with a spatially weighted kernel, then usually embed the representation into a mean shift framework for target state inference [70, 51, 71]. Mean-shift is an iterative process for locating the maxima of a density function [72]. Another example is the subspace learning-based generative model trackers. In these methods, the target appearance are usually represented by a set of basis subspace templates, therefore they focus on employing various subspace modelling techniques for visual representation, from the linear subspace models [73, 74] to non-linear or multi subspace models [75, 76]. Since generative trackers mainly focus on fitting the data from the object class while ignoring the background influence, they tend to be distracted by regions in the background with similar appearance.

Over the past few years, discriminative approaches (tracking-by-detection framework) have become one of the successful paradigms for object tracking. Compared to earlier generative methods, they have achieved superior performances [77, 78, 5]. As these methods share a lot in common with object detection, they have been termed *tracking by detection*. The tracking task is solved as a classification problem, they usually focus on training a model via a discriminative classifier to separate the target from the background. Numerous classifiers have been studies for visual tracking. Some of the representative examples are boosting-based [79, 80, 81]. The goal of boosting is to build a strong classifier by selecting multiple weak classifiers. Besides, large amount of research also chooses to use SVM [82, 5, 83], logistic regression [84, 85] rather than boosting, since it is more flexible to represent targets using kernels, and is robust to noise. Since the discriminative classifier is responsible for returning the confidence whether a candidate belongs to the target or not, it is usually believed to be the most important component of a tracking framework. In [85], the authors decompose a tracking framework and diagnose the effect of each individual component, the findings contradict general intuition in certain way. The experiments showed that observation model plays a crucial role when combined with a weak visual representation, which is obvious and predictable. However, when the visual representation is strong enough, the performance gap between different observation

models decreases dramatically. The findings provide an interesting insight into the research in this field. Instead of only focusing on the study of observation model, other components equally deserve the attention when it comes to the design of a tracking framework.

The conventional machine learning techniques had exploited shallow structure models, one of their common characteristics is the simple one layer architecture which converts the raw input or features into the task-specific feature space. Shallow models have been proven effective for various well constrained tasks, but they are restricted by their relatively limited representational and modelling power required by complicated real-world applications. These concerns give impetus to the development of building deep models for extracting complex representations from rich input data. deep learning based generative models have attracted more and more attention. They are often associated with unsupervised feature learning, since the labels of the data are not of concern. The models are intended to find the correlation of the observed data for pattern analysis [86]. One of the most common unsupervised deep models is autoencoders [87]. It is a non-linear feature extraction method, which is used for effective encoding learning or dimensionality reduction.

CNNs combine the feature representation and discriminative observation model into a unified framework. Compared to deep generative models, deep discriminative models are intended to characterize the posterior distributions of classes conditioned on the observed data. Historically generative models are used to facilitate the training of deep networks especially when the training data is limited, purely discriminative training from random initial weights is now proven to work very well with large amount of training data. CNNs driven by large scale training data and the rapid development of computation resources, initially showed their outstanding ability for image classification [64, 88, 89], then are applied to other visual tasks such as semantic segmentation [90, 91, 92], object detection [93, 94, 95] and many others [96, 97, 98]. Despite the huge success of CNNs, visual tracking application has been less affected by this trend due to the lack of training data for visual processing. Several work [38, 99, 100] have been alleviating this problem by using pre-trained CNN on other tasks (mainly image classification) as a black box feature extractor. Although off-the-shelf deep features may be sufficient for general visual representation, they cannot make up for the fundamental discrepancies between different tasks or bridge the gap between offline learning and online tracking. Motivated by this, [101, 102, 39, 40] have proposed tracking frameworks to fully explore the representation power of CNNs for tracking specific task.

2.1.3 Motion Model

Motion model, also as known as dynamic model, describes the dynamic behaviour of the target states, which usually consist the position and other localisation features of the target, such as velocity and acceleration, etc. From the data fusion point of view, the task of visual tracking is to estimate the state of the target based on the motion model by taking all available observations into account. Based on the characteristics of the tracker, various motion models have been developed, among which the most common motion models are constant velocity or constant acceleration.

The commonly used model especially for discriminative trackers is a dense sampling

method, which assumes the motion model of the target as constant velocity. Therefore, samples are drawn from a region around the location at previous time with Uniform distribution. This model does not maintain a distribution of the target location at every frame, so can only handle simple situations when the inter-frame object motion is small and smooth. Since tracking can be formatted as a dynamic state estimation problem, more advanced motion models can be used as a prior in the framework of Kalman filter or Particle filter.

Kalman filter is an estimator used to estimate the state of a linear dynamic system. The measurements are often the frame at current time, which are linear functions of the states but are polluted by additive Gaussion white noise. And the motion model used are also linear models such as constant velocity or acceleration. It performs in a form of feedback control: the filter estimates the state regularly and improves whenever the (noisy) measurement comes. As such, the whole process falls into two steps: the *Predict* step and the *Correct* step. The *Predict* step projects the current state ahead of time using previous measurements based on the motion model, and the *Correct* step adjusts the projected state by incorporating the current new measurement. Kalman filtering methods assume that the state posterior density is Gaussian, which is often violated in the real tracking problems. Particle filters, on the other hand, are sequential Monte Carlo methods which can model non-linear and non-Gaussian state space and generalize the Kalman filtering method. They approximate the posterior density function by a set of random particles with assigned weights, the weights represent the probability of that particle being drawn from the probability density function. Compared to dense sampling methods, Particle filtering methods are relatively computational efficient and are insensitive to local minimum, therefore various particle filter based trackers [74, 103, 85] have been proposed and achieved good performances.

2.1.4 Model Adaptor

The appearance of the tracking target will eventually change over time. Different from object detection problem, there is usually only one reliable example for visual tracking (e.g. from the first frame), and in some situations, like surgery, it is inevitable that the original target definition and model will change over the time due to dynamic effects like deformation, illumination changes, some and so on. Early tracking approached relied on fixed models which tend to drift away with significant appearance changes and cause the tracker to gradually lose the target and estimation the motion of irrelevant image regions. Adaptive models which evolves with appearance changes are key element for robust tracking. Model adaptor strategies should consider both when and how the observation model is updated, striking a balance between preventing drift and adapting to new but potentially noisy models during tracking.

Different adaptor strategies for the generative model LK method have been compared [104], where the main line of thinking was to update the current template by referencing the original template (starting model) to correct drift. This *template update with drift correction* strategy is also generalized to more complex observation models such as Active Appearance Models, which can improve the tracking robustness. To better account for the appearance changes, other update strategies have also been proposed in forms such as online boosting [79], incremental Principal Component Analysis (PCA) [74] and online multi-lifespan dictionary

learning [105].

For discriminative models, the focus is how to effectively draw positive and negative samples for training classifiers online. The method commonly used is to consider the current location as one positive example or even draw multiple positives from a tight neighbourhood, and then the negative examples are sampled around a larger neighbourhood. The concern is that if the tracking location is not precise, the appearance is updated with a sub-optimal positive example, which overtime also drifts to background. Additionally, multiple positives may pollute the appearance model with negative information causing the model being less discriminative. A different approach is to use semi-supervised training, in which the online classifier is trained with labelled examples from the first frame and unlabelled data from the subsequent frames [80]. This particularly suits the scenario where the target is completely out of view but is dependent on the initial labels. Multiple Instance Learning can potentially alleviate the inherent ambiguities of labelled data [78] and so can structured output prediction, which directly predicts the change of target location between frames [5]. Kalal et al. [3] proposed Tracking-learning-detection framework, in which the model is updated through P-N learning [106]. The learning process exploited the constraints of the structured unlabelled data to iteratively improve the performance of the classifier in a bootstrapping fashion. Fusion methods are also proposed. Yu et al. [107] co-trained based method to online update a hybrid discriminative generative model. Santner et al. [108] combined a simple non-adaptive template model, a moderately adaptive online random forest and a optical flow based mean-shift tracker as a highly adaptive element in a cascade to cope with various appearance changes.

2.2 Computer Vision for Image Guidance in Minimally Invasive Surgery

Methods from computer vision play a critical role in image-guided navigation, coupled with the development of advanced technologies such as HD endoscopy, intra-operative medical imaging and robotics introduced in MIS. Methods such as tissue surface tracking, registration or reconstruction, motion tracking, instrument detection, tracking and pose estimation have been applied to enhance the visualization and facilitate the surgeons during procedures. In this section, we will focus primarily on the review of literature specifically applied to endoscopic images as a means to enable navigation and to improve visual understanding of the surgical environment.

2.2.1 Tissue Surface Tracking and Recovery

Information about the shape, motion and deformation of the tissue surface is critical measurement for image guidance and navigation. Different tissue surface representations have been used in vision based methods to estimate and recover the information from acquired endoscopic images. Groger and Ortmaier et al. [109, 110, 111] exploited sparse natural epicardial landmarks on the tissue surface to predict short term 2D motion of the heart in beating heart surgery. Stoyanov et al. [112, 113] extended the idea in 3D, the authors used a pre-calibrated stereo laparoscope, matched a set of salient features from stereo images to infer their 3D position, and then estimated the 3D motion of the features through iterative LK tracking method. For sparse region-based methods, the quality of detected salient regions is crucial. In MIS

images, it can be difficult to detect salient regions and match corresponding regions in stereo setup due to specular reflections, homogeneous tissue texture or illumination changes. Different methods for identifying salient regions are compared in [114]. For feature matching, different feature descriptors were investigated in [115], a probabilistic framework was then proposed to select and fuse the most discriminative descriptors for MIS deformation estimation.

Salient feature tracking can only provide sparse motion from individual points. To infer dense motion of the tissue surface, studies have been proposed by incorporating geometric tissue models into tracking as motion model. In [110], since it is hard to represent the nonlinear soft-tissue motion, the motion patterns of the landmarks were approximated by a linear model, and a simple affine motion model was used to describe the global motion of the heart surface. In following papers, 3D motion of the tissue surface was tracked using stereo with more advanced models, such as B-spline [116] and piece-wise bilinear maps (PBM) [117, 118]. Richa et al. [119, 120, 121] used a Thin-Plate Splines (TPS) model, which does not require explicit stereo rectification or matching compared to previous models. Those deformable models provide motion constraints and provides dense motion predictions through interpolation. In [122], TPS model was also used to align multiple overlapping retina maps. Ye et al. [123] proposed to recover quasi-dense 3D structure based on matched 3D feature correspondences across time. The density of the initial set of sparse reliable features was gradually increased by applying local search and feature propagation based on affine consistency of anisotropic regions.

In summary, most tissue surface tracking is based on tracking-by-model-fitting. The region of interest (ROI) of tissue is generally represented by features in feature space or raw pixel values in image space, since global tissue motion is too complicated to parametrize, it is usually simplified to models with small number of parameters, ranging from translation, affine models to non-rigid ones. Then to get dense motion of the tissue surface, generative methods such as iterative optimization are used to find the most similar target appearance based on the motion model. The reasons discriminative tracking methods are rarely used for tissue tracking are that unlike tracking general objects, it is difficult to model the background in surgical environment, there is no clear boundary for the ROI, and to recover non-rigid dense soft tissue motion, it requires more complicated motion model. But factors in the complex surgical environment such as occlusion, illumination variations and abrupt appearance changes also make the tracking challenging for tracking by model fitting methods.

2.2.2 Surgical Instrument Detection and Tracking

Surgical instrument localization is another important task in robotic instrument control for RMIS. Information from different sources has been used for instrument tracking shown in Figure 2.4. Typically colour, gradient or texture is employed to represent the appearance model, instrument shape can be simplified or explored using a prior model to confine the search space. In addition, cues such as robotic kinematics can also be used as external constraints.

In [126], a robotic laparoscope positioner was developed to replace the surgical assistant, which eliminates the hand trembling. To simplify the control of the robotic scope positioner, a novel automatic tracking method was proposed to localize the instrument, and the output is used to automate the process of scope maneuvering. To track the instrument, pixels were classified

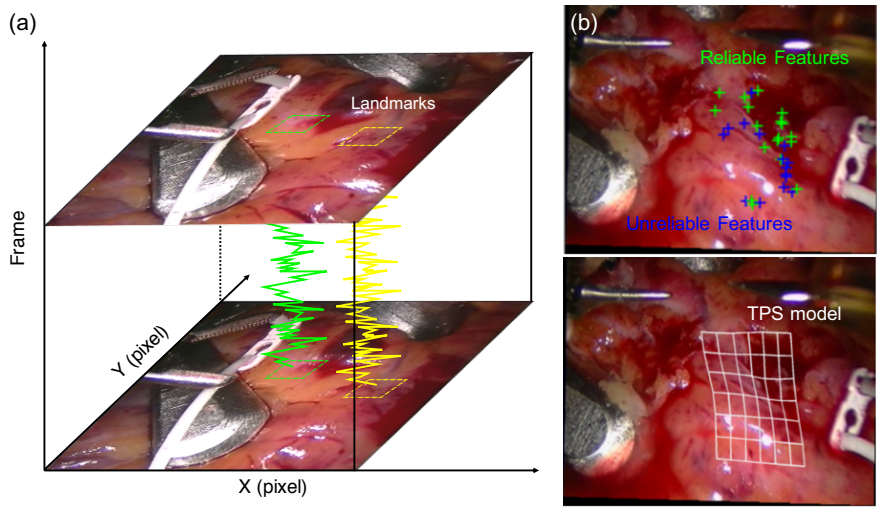


Figure 2.3: Tissue surface tracking methods: (a) Salient region motion tracking in open heart surgery, and sparse motion is obtained [124]; (b) The tissue surface motion is represented bt TPS model, the motion of unreliable features (marked in blue) can be interpolated using the motion of reliable features (marked in green) [121].

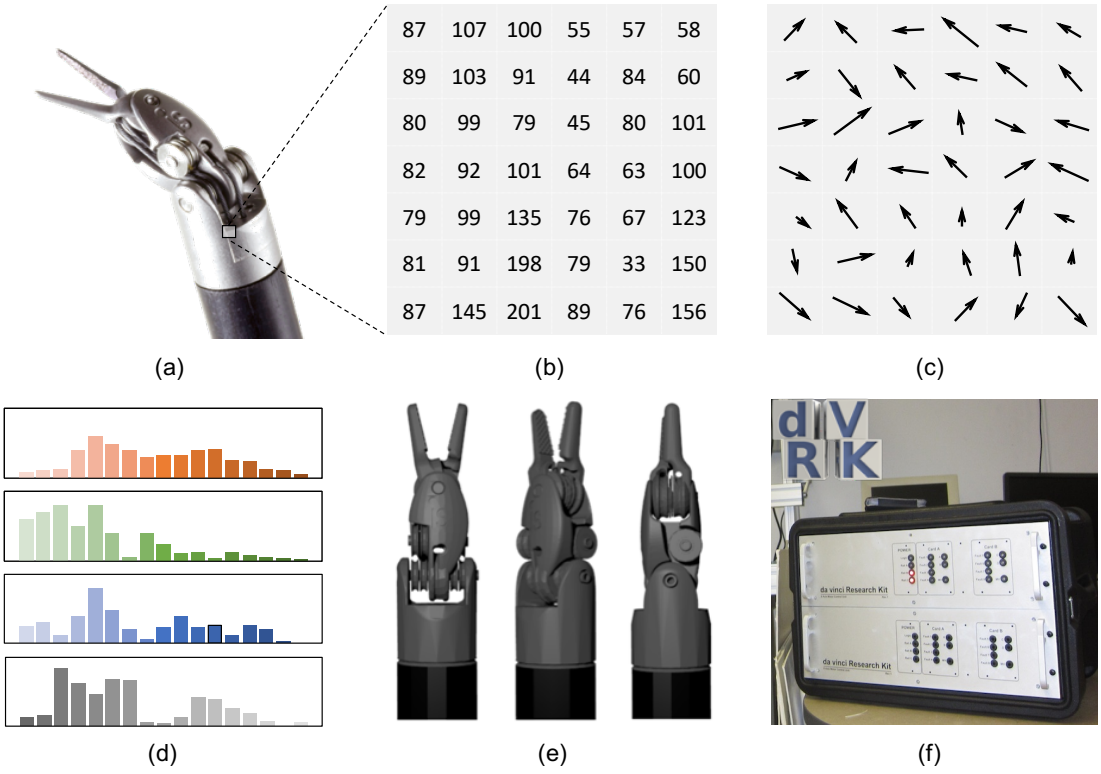


Figure 2.4: Common information used for instrument detection or tracking methods: (a) Articulated da Vinci Instrument; (b) Raw image pixel information; (c) Gradient information; (d) Histogram of colour or gradient; (e) Instrument CAD model prior; (f) Kinematic information retrieved from the da Vinci Research Kit (dVRK) [125].

based on the colour of the instrument and surrounding organs, the motion of the instrument was then predicted by shape analysis and temporal filtering. This established the basic concept for instrument tracking in the following years.

To make the colour of the instrument more distinctive, artificial colour markers were designed and mounted to the instrument [127, 128]. Zhang et al. [129] proposed to use multiple black strip markers for monochrome processing, instead of colour markers for simpler monochrome processing. Specular reflection does not affect or even is beneficial to the detection of markers, since intensity difference between the reflected object and the marker is even larger. Zhang et al.[130] proposed a new cylindrical marker design which consists of different patterns of circular dots and chessboard vertices.

Although attaching markers on instrument makes the detection more robust and simple, the idea of modifying instruments is usually avoided since it changes the surgical procedure. Also, artificial markers may introduce inconvenience, such as biological hazard or retrofittable difficulty. In [131], a flexible shape model was proposed to represent the motion model, and particle filter was used to propagate the model state using colour measurement.

In [132], not only colour, gradient was also incorporated to detect the 2D location and the edge of the instrument, and then 3D orientation and location were inferred by modelling the instrument as a cylinder. Reiter et al. [133] proposed to learn the appearance of the instrument online by combining multiple features, and explores new areas as the instrument moves in or out of view. To handle instrument re-detection when it enters or disappearance from scene, data-driven detection is incorporated in tracking framework. Pezzementi et al. [134] was the first to track articulated instrument with known geometric structure. A colour appearance model was trained offline using Gaussian Mixture Models (GMM) with manual pixel-wise segmentations. Since the geometric model of instrument was known, the 3D model was projected and aligned with the 2D class probability image generated by the classifier through optimization. The method takes advantage of machine learning methods to create a generative appearance model and align known 2D geometry model projections with measurements. Allan et al. [135] learns a random forest to classify instrument in pixel-wise fashion, then the binary classification output was used to estimate the pose of a prior 3D instrument model through optimization within a level set framework. Following the work, in [136] the work was improved by combining constraints from feature points, temporal motion model with stereo setup. In [137], instead of binary silhouette, separate part appearance models were used to align the prior model with low level optical flow constraints. Alsheakhali et al. [138] proposed to use probabilistic Hough Transform to optimize the location by fitting on the colour histogram-based instrument segmentation. In [139], kinematics obtained from robot was also employed for robust instrument tracking. Part-based templates were generated online based on prior 3D CAD model for instrument detection, 3D kinematic data are fused with the 2D detection to estimate 3D pose of the instrument.

Recently, the widespread success of deep learning techniques has led to advanced approaches for instrument classification, segmentation and instrument detection (Figure 2.5). EndoNet [140] was designed to detect instrument presence jointly with phase recognition in laparoscopic surgeries. For training the model, a new instrument presence dataset was proposed for cholecystectomy procedures. In [141], multi-label imbalance problem was addressed for surgical instrument classification for endoscopic video stream. Mishra [142] proposed a tool presence detection framework by using CNNs as feature extractor and a Long Short-Term

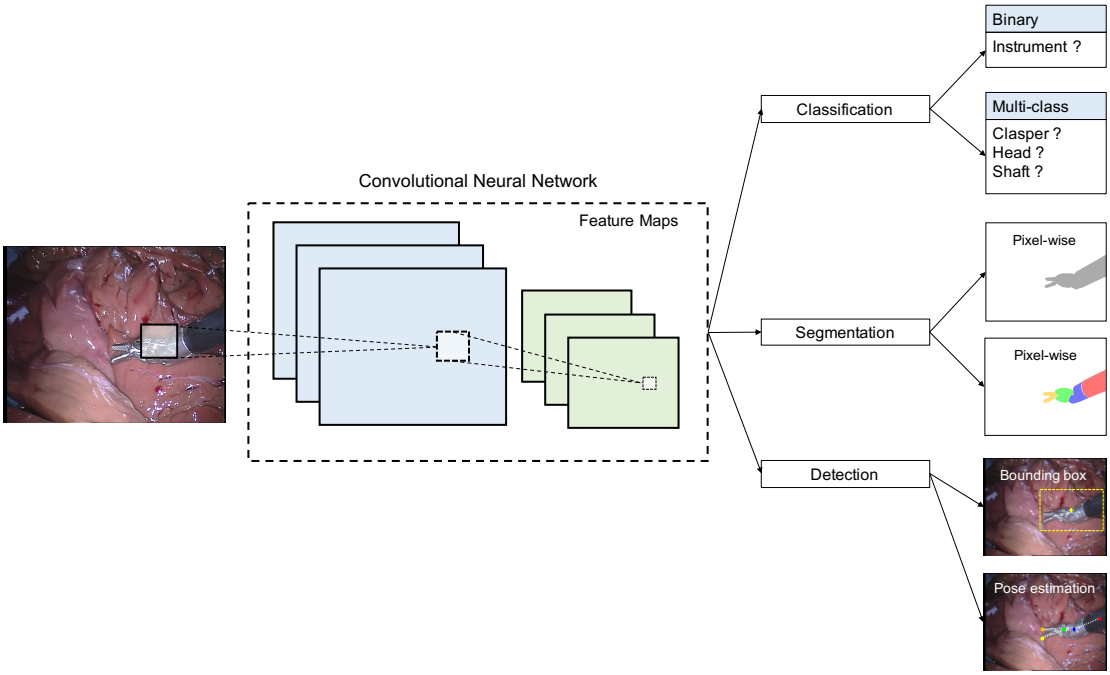


Figure 2.5: Deep learning based instrument applications ranging from classification, segmentation, detection, etc.

Memory network as temporal information encoder. In [143], the tip point of the instrument was detected using a well-trained neural network, and the shaft was depicted by line features using Random Sample Consensus (RANSAC) scheme. Choi et al. [144] constructed new location annotations for the EndoNet dataset, and proposed real-time CNN model for localizing various instruments during laparoscopic surgery.

In addition to endoscopic or laparoscopic surgery, instruments are tracked from stereo microscope for retinal microsurgery. Richa et al. [145] proposed to use a novel similarity measure based on weighted mutual information for gradient-based iterative optimization. The similarity metric has the advantage of being robust to illumination variations and partial occlusions. In [146, 147] a unified framework by combining detection and tracking was proposed as a Bayesian estimation problem. And in [148] an offline detector was trained and coupled with a gradient-based tracker to produce position estimates. Rieka et al. [149] modelled the problem as two different tasks, tracking and pose estimation, both of which employed online random forests. The instrument was first tracked and then parts were estimated in the tracked bounding box. The work was extended to colour space [150] and a failure detection module was added to the framework for instrument re-identification [151].

In [152], retinal instruments are modelled using Conditional Random Field (CRF), it relies on deep neural network to classify instrument parts, and the potentials of CRF are learnt to estimate not only the location but also the orientation of the instrument. Probst et al. [153] proposed an automatic pipeline for tool localization in a stereo microscope. A retinal instrument detection network was designed to localise different parts of the instrument, the 2D detected coordinates along with 3D coordinates of the instrument obtained from the robot were used to calibrate the microscope. Laina et al. [154] proposed a multi-task CNN model for concurrent

semantic segmentation and multi-part localization for retinal instruments.

2.2.3 Major Challenges for Visual Tracking in Surgical Applications

The challenges of the visual tracking tasks in surgical applications starts with the image acquisition system. Image quality (such as image resolution, lens deformation, etc), synchronization and acquisition speed are essentially hardware problems. Specifically for tissue surface tracking, Ginhoux et al. [155] pointed out that the heart motion has very fast transients and with a slow acquisition rate, information loss due to aliasing is not negligible. They also suggested an acquisition speed not smaller than 100 Hz for compensating the heart motion. Current available image acquisition hardware offers limited acquisition rates, motion blur induced by fast motion is therefore one of the challenges. In Figure 2.6, we show some major challenges in surgical tracking applications. Besides, certain regions of soft-tissue surface or instruments do not provide distinguish or stable feature or texture information. While using artificial markers to add distinctive features is not practical in most cases. Another important source of challenges for the tracking procedure is lighting changes. Since the workspace in MIS is often restricted, the light source illuminates unevenly the operating site. Associated with the physiological motion, the brightness constancy assumption on which various tracking methods are based upon is violated, making the visual tracking task complicated. Also, direct reflections of the light source on the glossy, wet-like surface or metal instruments give rise to specular reflections. These unreliable specular reflections may be interpreted as texture information by the tracking algorithm. In the dynamic surgical procedure, liquids and smoke which are present at the operating site often disturb visual tracking. It is also expected that as the surgeon manipulates the tissue its appearance significantly changes. Besides, the manipulation of surgical instruments can also cause tissue region or instrument occlusions. Due to the unpredictable and complicated nature of the surgical procedure, a robust tracking framework needs high-level visual understanding towards the surgical scene to eliminate or decrease the effects of all the above factors.

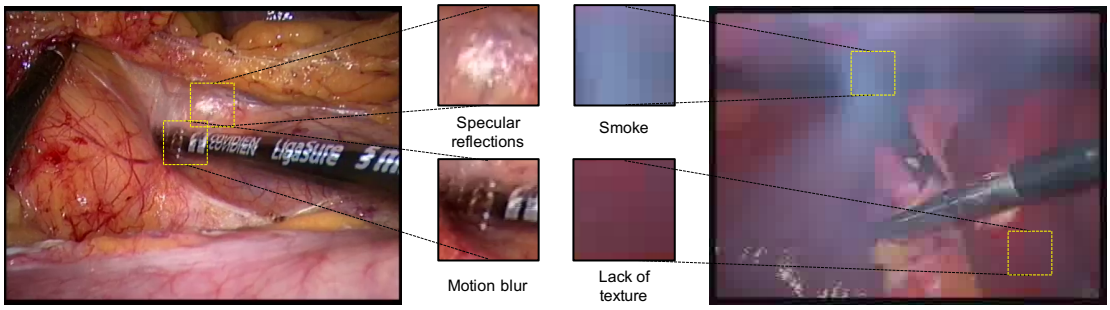


Figure 2.6: Challenges for visual tracking tasks in surgical applications, such as specular reflections, motion blur, smoke, lack of texture, etc.

2.3 Discussion

Visual tracking is a key research area within computer vision and it is closely related to other vision problems like object detection, pose estimation, structure reconstruction and semantic segmentation. This chapter has concisely introduced the key components of a visual tracking framework because a thorough review of the field is intangible with numerous new papers

reporting algorithm advances throughout the year. Depending on the specific goals and requirements of individual research application, the tracking approaches being used may differ from each other, and have their own advantages and disadvantage. Until now, there is no universal strategy or solution to tackle the problem. Although the architecture of a complete automatic tracking systems is still an open problem, in principle we can still decompose a tracking system into functional components: visual representation, observation model, motion model and model adaptor. In the chapter, we cover their roles, developments, contributions and state-of-the-art research in the literature. We then reviewed how tracking techniques have been used for the estimation the motion of structures visible in MIS video acquired through an endoscope. Despite being a much less active field than general computer vision, quite a number of new techniques are looking as solving the tracking problem in computer assisted endoscopy but due to the challenging nature of the task robust systems ready for practical application and clinical translation are still lacking. In this remainder of this thesis we focus on developing such methods.

Chapter 3

Non-rigid Deformation Tracking for Soft Tissue Surface

3.1 Introduction

Medical image computing and surgical vision can play an important role towards improving the surgeon's operating capabilities in highly dynamic anatomical regions where tissue motion can complicate surgical dexterity and impede image-guidance or intra-operative imaging [28, 156]. In MIS, recovering *in vivo* tissue deformation in real-time by using endoscopic images has been explored predominantly for deploying robotic motion stabilization [117]. While both 2D and 3D tracking methodologies have been reported the problem of robustly tracking tissues with poor texture characteristics remains a challenging task due to the illumination complexity and variation, specular highlights and occlusions from the surgical instruments [157].

Early work on tracking tissue motion in endoscopic video focused on the use of feature-based methods in order to achieve real-time performance [158, 159]. More recently robust feature driven techniques have been developed and reported to achieve robust and long-term tracking invariant to difficult transformations [160, 161, 162]. The limitation of these approaches is that a dense region of the tissue is not recovered and rather single points of interest are detected and tracked which these can be isolated in specific regions. On the other hand, dense intensity-based methods have been reported where the tissue surface is modelled as a geometric mesh, for example, using a TPS or Free-form deformation technique. Tracking is performed over the entire space covered by this model. Richa *et al.* [119] employed a TPS model to estimate the heart surface deformation using multiple visual techniques to increase robustness and spatial resolution. Braux-Zin *et al.* [163] introduced a new model of non-rigid surface registration to merge feature and intensity-based costs in a pyramidal variational approach, however, the model fails in the presence of illumination variations. Besides, additional specialized hardware can also be used for soft-tissue reconstruction [164].

3.2 Non-rigid Deformation Tracking

We model the tracked tissue surface as a geometric mesh model with regularization constrains which can describe the deformable tissue motion. We combine the advantages of both feature and intensity information to track the tissue surface robustly and in difficult conditions with poor illumination. The energy cost function we optimize incorporates terms for feature correspon-

dence energy and also for intensity energy. During tracking, the locations of the mesh vertices are updated to minimize energy functions with regard to the feature correspondence alignment, dense image residuals and also illumination variations. We report encouraging results and compare our algorithm with earlier works to show that performance is enhanced and the method can cope with large motions due to the feature components while also handling regions of poor textural information through the use of illumination compensated appearance. We also show preliminary results applying our method to multispectral images, where the signal can be low and difficult for tracking algorithms.

3.3 Geometric Mesh Model

In this work, we use the mesh model proposed by Pilet *et al.* [63]. As shown in Figure 3.1, the non-rigid tissue surface M to be tracked is modelled as a 2D triangular geometric mesh with N vertices. $\mathbf{v_i} = (v_{ix}, v_{iy})$ represents the pixel location of the i^{th} triangle vertex of the mesh. Let $\mathbf{V_x} \in R^N$ and $\mathbf{V_y} \in R^N$ be the vector of stacked vertex coordinates of the x-axis and y-axis respectively. To represent the shape and motion of M, we define a state vector $\mathbf{S} = (\mathbf{V_x}; \mathbf{V_y}) \in R^{2N}$.

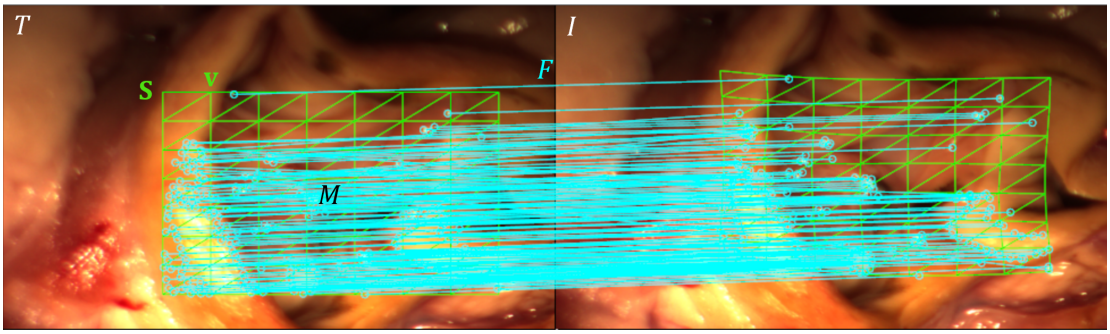


Figure 3.1: The left and right images are the template image T and the input image I respectively. F is the set of feature correspondence obtained using feature matching algorithm (shown as cyan). The tissue surface M is modelled as a triangular mesh model (shown as green), so the deformation and motion of the surface M are controlled by the state vector $\mathbf{S}$ consisting of the mesh vertices $\mathbf{v}$.

Any point $\mathbf{p}$ within M can be located via the warping function $\mathbf{W}(\mathbf{p}; \mathbf{S})$ by using its barycentric coordinate (b_i, b_j, b_k) and the vertices of the triangle it lies within $(\mathbf{v_i}, \mathbf{v_j}, \mathbf{v_k})$, where the triplets (i, j, k) represent the triangle vertex indices. Barycentric vector $\mathbf{b} \in R^N$ only contains non-zero elements at index (i, j, k).

$$\mathbf{W}(\mathbf{p}; \mathbf{S}) = \begin{bmatrix} \mathbf{b^T} & \mathbf{0} \\ \mathbf{0} & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{V_x} \\ \mathbf{V_y} \end{bmatrix} = \mathbf{BS} \tag{3.1}$$

where $b_i + b_j + b_k = 1$. Then the task of surface tracking is to estimate the state vector $\mathbf{S}$ of the mesh through the whole image sequence.

3.4 Feature-based Tracking

One advantage of our framework is that it can cope with any kind of feature detection and tracking method as originally presented by [114]. In this study we used the SURF descriptor [59] implemented in the OpenCV library [165] to obtain feature correspondences.

Given a set of feature correspondences, to estimate the deformation of M with tissue displacement, we minimize an energy function subject to $\mathbf{S}$ in the following equation:

$$\varepsilon_F(\mathbf{S}) = \lambda\varepsilon_R(\mathbf{S}) + \varepsilon_C(\mathbf{S}) \tag{3.2}$$

where ε_R represents the regularization energy term, and the residual term ε_C is a feature correspondence measure and λ controls the regularization influence [63].

No matter what kind of feature matching algorithm is used, it is usually inevitable to avoid erroneous correspondences, which we consider as outliers. The presence of outliers can severely affect deformation estimation, for instance by breaking the mesh topology. Therefore, the regularization energy ε_R is used to prevent the mesh model from overfitting the data. The mesh model can be considered as a set of hexagons, one of which is shown in green in Figure 3.2.

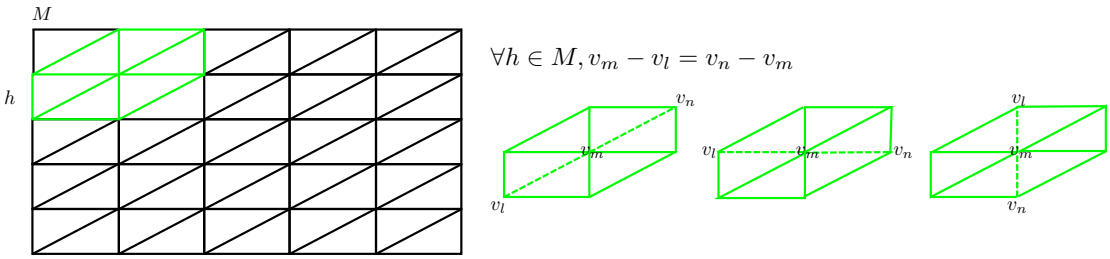


Figure 3.2: A hexagonal element h in the undeformed mesh model (shown in green). The distance between co-linear vertices is equal under certain types of hexagon motion.

For each hexagon triplets (l, m, n) in the undeformed mesh model, the distances between all the co-linear vertices are equal respectively. This property can be used to preserve the regularity of the mesh. We separate the coordinates of the vertices along x and y axis, and therefore the regularization energy term is defined as in [166]:

$$\begin{aligned}\varepsilon_R(\mathbf{S}) &= \frac{1}{2}\sum_{(l,m,n)\in E}(v_{lx} - 2v_{mx} + v_{nx})^2 + (v_{ly} - 2v_{my} + v_{ny})^2 \\ &= \frac{1}{2}\sum_{(l,m,n)\in E}\sum_{i=x,y}\begin{bmatrix} v_{li} \\ v_{mi} \\ v_{ni}\end{bmatrix}^T\begin{bmatrix} 1 \\ -2 \\ 1\end{bmatrix}\begin{bmatrix} 1 \\ -2 \\ 1\end{bmatrix}^T\begin{bmatrix} v_{li} \\ v_{mi} \\ v_{ni}\end{bmatrix}\end{aligned} \tag{3.3}$$

where E is composed of all the index triplets (l, m, n) for the co-linear vertices. For convenience, this can be formulated in the matrix form:

$$\begin{aligned}\varepsilon_R(\mathbf{S}) &= \frac{1}{2}(\mathbf{V_x^T K V_x} + \mathbf{V_y^T K V_y}) \\ &= \frac{1}{2}\mathbf{S^T}\begin{bmatrix}\mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}\end{bmatrix}\mathbf{S} = \frac{1}{2}\mathbf{S^T R S}\end{aligned} \tag{3.4}$$

where $\mathbf{R} \in R^{2N\times 2N}$ is a sparse and regular matrix which can be determined by the set of triplets E.

Let $\mathbf{c_t} = (u, v)^T$ represent the location of one feature in the frame t, and F be the set of all correspondences we obtained after feature matching. The feature correspondence energy is

defined as:

$$\varepsilon_C(\mathbf{S}) = \sum_{\mathbf{c}\in F} \rho(\|\mathbf{c_t} - \mathbf{W}(\mathbf{c_{t-1}}; \mathbf{S})\|, r) \tag{3.5}$$

where ρ is a robust estimator, and r is the confidence radius. The choice of the robust estimator is crucial for decreasing the effect of outliers. Various robust functions exist [163, 63, 167], in this work we follow the estimator and optimization method proposed by Zhu *et al.* [167]. Based on the modified finite Newton method [168, 169], Zhu *et al.* proposed the Progressively Finite Newton (PFN) method [167], in which the robust estimator $\rho(\delta, r)$ uses a coarse-to-fine scheme. The initial value of r is set to 500 and is progressively reduced at a constant rate. The optimization process stops when the confidence radius r reduces to one pixel which is close to the expected precision. Because the method only needs one Newton step for each r to achieve convergence, the whole process can be solved in a fixed number of steps.

3.5 Deformable Lucas-Kanade Method

Feature-based tracking is fast and can handle large displacements, but it has limitations because of the sparse motion field and the reliance on salient image texture. In [167], the authors employed a deformable Lucas-Kanade (DLK) method, which is a deformable variation of the intensity-based LK method [37]. The energy function subject to $\mathbf{S}$ is defined as:

$$\varepsilon_I(\mathbf{S}) = \eta\varepsilon_R(\mathbf{S}) + \varepsilon_{SSD}(\mathbf{S}) \tag{3.6}$$

where ε_R represents the regularization energy term and η controls the regularization influence. The residual intensity energy term ε_{SSD} uses the SSD as a similarity measure between the template image T and the input image I, here we use the inverse compositional form of the LK method which minimizes:

$$\varepsilon_{SSD}(\mathbf{S}) = \frac{1}{2}\sum_{\mathbf{p}} \left[T(\mathbf{W}(\mathbf{p};\mathbf{\Delta S})) - I(\mathbf{W}(\mathbf{p};\mathbf{S}))\right]^2 \tag{3.7}$$

with respect to $\mathbf{\Delta S}$ for each pixel $\mathbf{p}$ in the ROI, and then the warp is updated:

$$\mathbf{W}(\mathbf{p};\mathbf{S}) \leftarrow \mathbf{W}(\mathbf{p};\mathbf{S}) \circ \mathbf{W}(\mathbf{p};\mathbf{\Delta S})^{-1} \tag{3.8}$$

where the incremental warp $\mathbf{W}(\mathbf{p};\mathbf{\Delta S})$ is inversed before composing with the previous estimate.

The SSD metric directly compares the illumination of every pixel $\mathbf{p}$ in the tracked area, which makes it quite sensitive to changes in lighting. Recently, a new similarity metric called the Sum of Conditional Variance (SCV) was introduced for multi-modal medical image registration [170], and Richa *et al.* [171] used it for visual tracking. It calculates the sum of conditional variances for images T and I. In MIS, compared to SSD, SCV is invariant to non-linear illumination variations. In this study, we improved the DLK method of Zhu *et al.* [167] by employing the SCV metric. Let $[0, d_T]$ and $[0, d_I]$ represent the intensity range of the template image T

and the input image I respectively, the intensity energy ε_{SCV} is defined as:

$$\varepsilon_{SCV}(\mathbf{S}) = \frac{1}{2}\sum_{\mathbf{p}}\left[T(\mathbf{W}(\mathbf{p};\mathbf{\Delta S})) - \hat{I}(\mathbf{W}(\mathbf{p};\mathbf{S}))\right]^2 \tag{3.9}$$

with the SCV image

$$\hat{I}(\mathbf{W}(\mathbf{p};\mathbf{S})) = \mathscr{E}(T(\mathbf{p})|I(\mathbf{W}(\mathbf{p};\mathbf{S}))) \tag{3.10}$$

where $\mathscr{E}(\cdot)$ is the expectation operator. This can be computed using the joint intensity distribution $P_{ij} \in R^{d_T \times d_I}$ between T and I:

$$P_{ij} = \frac{1}{N_{\mathbf{p}}}\sum_{\mathbf{p}}\Phi\big(T(\mathbf{p}) - i\big)\Phi\big(I(\mathbf{W}(\mathbf{p};\mathbf{S})) - j\big) \tag{3.11}$$

where $N_{\mathbf{p}}$ represents the number of pixels. $i \in [0, d_T]$ and $j \in [0, d_I]$ represent the discrete pixel intensity numbers that the template image T and the input image I have, respectively. $\Phi(x) = 1$ if and only if $x = 0$. Therefore, each element of the joint distribution P_{ij} represents the probability of the intensity concurrence (for $T(\mathbf{p}) = i$ and $I(\mathbf{W}(\mathbf{p};\mathbf{S})) = j$) for a given pixel $\mathbf{p}$. The conditional expectation can then be computed as:

$$\mathscr{E}(T(\mathbf{p})|I(\mathbf{W}(\mathbf{p};\mathbf{S}))) = \frac{\sum_i i \cdot P_{ij}(i, I(\mathbf{W}(\mathbf{p};\mathbf{S}))}{\sum_i P_{ij}(i, I(\mathbf{W}(\mathbf{p};\mathbf{S})))} \tag{3.12}$$

During tracking, the SCV image $\hat{I}(\mathbf{W}(\mathbf{p};\mathbf{S}))$ is computed only once for every input image I [145]. From above equations, we can see that the mapping of image intensities between the reference image T and the input image I is considered as a mapping function with assumption that parts of the target with the same intensity holds equal reflectance properties. Compared to the assumption that the whole target having the same reflectance properties, this weaker assumption enables SCV to cope with a larger variety of illumination variations. Then the optimization can be processed like the standard procedure for the LK method. In Figure 3.3, we displayed two multispectral images under different wavelengths, which show dramatic illumination variations. After SCV illumination mapping, the ROI of the input frame shares similar illumination condition with the template frame.

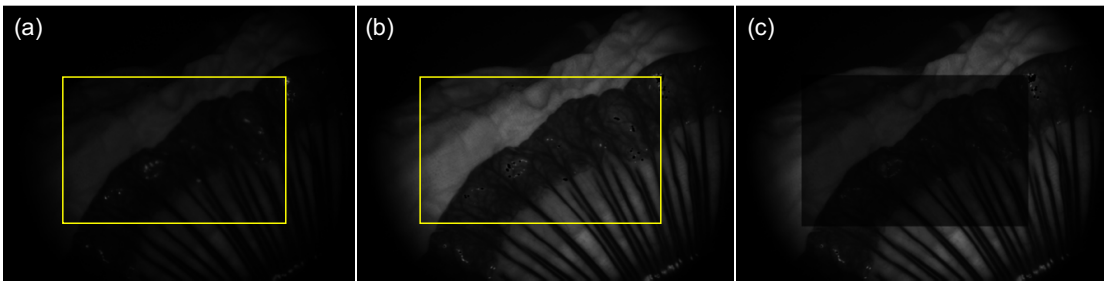


Figure 3.3: SCV illumination mapping example: (a) Template frame with ROI (bounding box in yellow); (b) Input frame with the same ROI (bounding box in yellow); (c) The ROI of the input frame is mapped to mimic the illumination condition of the template image using the SCV metric.

Performing a first-order Taylor expansion on $\varepsilon_{SSD}(\mathbf{S})$ in Equation 3.6 gives:

$$\eta\frac{1}{2}(\mathbf{S}+\mathbf{\Delta S})^{\mathbf{T}}\mathbf{R}(\mathbf{S}+\mathbf{\Delta S})+\frac{1}{2}\sum_{\mathbf{p}}\left[T(\mathbf{W}(\mathbf{p};\mathbf{0}))+\boldsymbol{\nabla}T\frac{\partial\mathbf{W}}{\partial\mathbf{S}}\mathbf{\Delta S}-I(\mathbf{W}(\mathbf{p};\mathbf{S}))\right]^2 \quad (3.13)$$

the above equation is a least squares problem, and assuming that $\mathbf{W}(\mathbf{p};\mathbf{0})$ is the identity warp, the partial derivative of the expression with respect to $\mathbf{\Delta S}$ is:

$$\eta\mathbf{R}(\mathbf{S}+\mathbf{\Delta S})+\sum_{\mathbf{p}}(\boldsymbol{\nabla}T\frac{\partial\mathbf{W}}{\partial\mathbf{S}})^T\left[T(\mathbf{p})+\boldsymbol{\nabla}T\frac{\partial\mathbf{W}}{\partial\mathbf{S}}\mathbf{\Delta S}-\hat{I}(\mathbf{W}(\mathbf{p};\mathbf{S}))\right] \quad (3.14)$$

and setting the derivative equal to zero, solving for $\mathbf{\Delta S}$ we have:

$$\mathbf{\Delta S}=\mathbf{H}^{-\mathbf{1}}\left[-\eta\mathbf{R}\mathbf{S}+\sum_{\mathbf{p}}(\boldsymbol{\nabla}T\frac{\partial\mathbf{W}}{\partial\mathbf{S}})^T(\hat{I}(\mathbf{W}(\mathbf{p};\mathbf{S}))-T(\mathbf{p}))\right] \quad (3.15)$$

with the Hessian matrix:

$$\mathbf{H}=\eta\mathbf{R}+\sum_{\mathbf{p}}(\boldsymbol{\nabla}T\frac{\partial\mathbf{W}}{\partial\mathbf{S}})^T(\boldsymbol{\nabla}T\frac{\partial\mathbf{W}}{\partial\mathbf{S}}) \quad (3.16)$$

Because any LK method is based on the assumption that the current estimate of the parameters is approximately correct. This means that by using ε_{SSD} or ε_{SCV} alone we cannot deal with significant displacement between frames [37]. We use the feature-based tracking result as initialization to fulfil this assumption and to lead the optimization toward correct convergence.

3.6 Template Updating

After an extended sequence of tracking, it is possible that the original template will not accurately represent the tracked surface due to physiological effects such as bleeding after instrument-tissue interactions. To avoid errors caused by the appearance changes we use a template updating strategy [104]. First, at every frame we update the template image to be the tracked region of the input image. In this way, the updated template could lessen the possible appearance difference between the original template and input image. At the same time small errors will accumulate during this process and cause the template to gradually drift away. Therefore, to correct the drift we keep the original template and align the updated template with it to estimate the final update. This two-step template update with drift correction strategy can avoid local minima during optimization and prevent the drifting problem. Additionally, specular reflections create strong image gradients which are salient and can bias feature detection and appearance-based tracking metrics. We use a combination of intensity thresholding and dilation operations to remove the highlights [172, 173] (see Figure 3.4).

3.7 Experiments and Results

3.7.1 Synthetic Data Experiments

As GT information for soft-tissue motion is not available during surgery, we used a custom simulation environment in Figure 3.5 to mimic the periodic deformation of the tissue surface

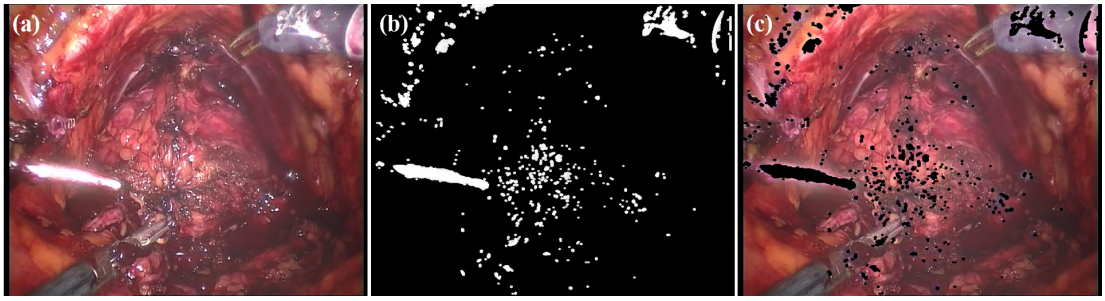


Figure 3.4: Specular highlight removal procedure before tracking: (a) Before highlight removal; (b) Highlight mask; (c) After highlight removal.

induced by the cardiac cycle and respiration [174]. The environment can generate synthetic image sequences by performing small but arbitrary rotations and translations to the pixels of one template image. To test the computational stability of the method, we also added different levels of Gaussian noise (5%, 10% and 20%) to the synthetic sequence. The percentage of the noise represents the percent ratio of the standard deviation of the white Gaussian noise versus the intensities of the whole image.

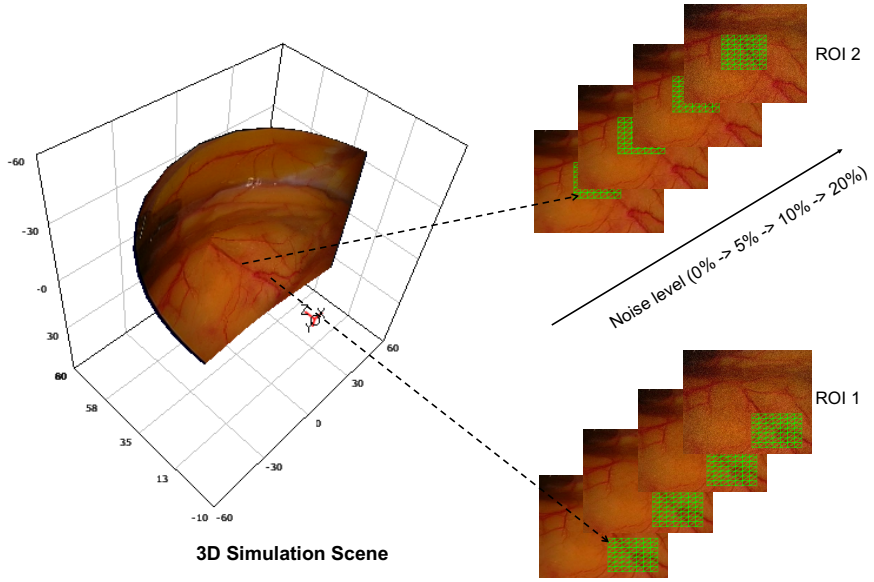


Figure 3.5: Simulation environment experiment setup for generating synthetic image sequences.

To quantitatively validate the performance of our algorithm, we tracked two ROIs as seen in Figure 3.5. The first ROI located near the right bottom corner deforms towards the centre during tracking, so the displacement is large between frames; the second ROI is located in the central area and is compressed during tracking.

Since we generated the whole synthetic sequence, we have the GT of the mesh vertices in each frame. We computed and compared the mean and standard deviation (STD) of the tracking error (pixel) compared to the GT with different methods: the feature-based PFN method, the modified intensity-based DLK using the SCV metric and our proposed hybrid PFNLK method.

The tracking results of the two ROIs with different noise levels through the sequence are shown in Figure 3.6. Since the displacement between frames of this sequence is too large, the

DLK method quickly loses track, while the PFN and our PFNLK method track fairly well. The PFNLK method outperforms all the other methods, but more obviously for ROI 2 than ROI 1.

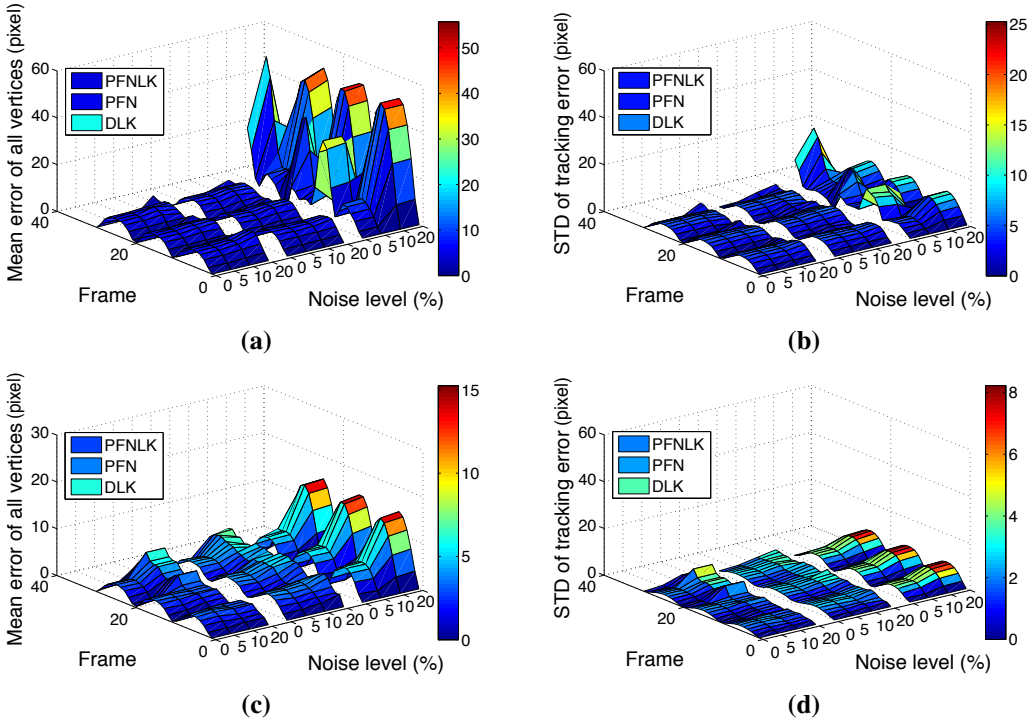


Figure 3.6: Synthetic experiment results: (a) Mean error for ROI 1; (b) The standard deviation of error for ROI 1; (c) Mean error for ROI 2; (d) The standard deviation of error for ROI 2.

3.7.2 *In Vivo* Data Experiments

For evaluating the potential clinical value of the proposed tracking algorithm, experiments on *in vivo* image sequences recorded at 25 fps using the da Vinci® surgical robotic platform (Intuitive Surgical, Inc.) have been conducted. Since the lack of GT is a problem for validating tracking performance with real surgical sequences, we used the modified forward-backward tracking methodology based on even-odd frames [175]. For a given sequence, the forward tracking is made on the even frames and the backward tracking is made on the odd frames. The assumption is that if a ROI is perfectly tracked, it should return to the initial location in the first frame. This is considered to be artificial GT for tracking methods. Compared to the original forward-backward tracking strategy [176], which tracked the ROI frame by frame as they move forward and backward to the beginning of the sequence, the backward tracking is decorrelated from the forward tracking by using different frames. In our experiments, we chose a robotic radical prostatectomy sequence, which is represented by $I = (I_0, I_1, I_2, \ldots, I_n)$. Then according to the above methodology, $FB = (I_0, I_2, I_4, \ldots, I_{n-2}, I_n, I_{n-1}, \ldots, I_3, I_1, I_0)$ is the corresponding modified forward-backward sequence where I_t is the frame t of the original sequence. As seen in Figure 3.7, the first frame (frame 0) is the same as the last frame (frame 50) in the FB sequence.

We tracked the same ROI with the three methods, and the tracking result can be seen in Figure 3.7. The DLK method loses track, but the PFN and the PFNLK methods track well. Until the last frame, the PFN method tracked back to near the original location, while the PFNLK

method tracked back to the initial location. Normalized Cross-Correlation (NCC) can also be

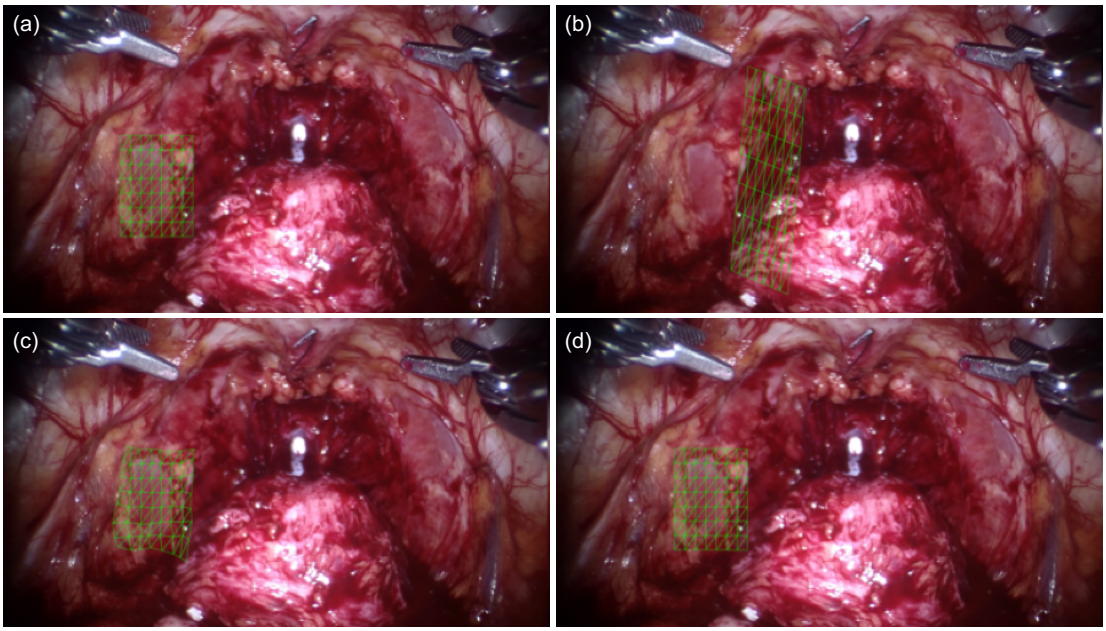
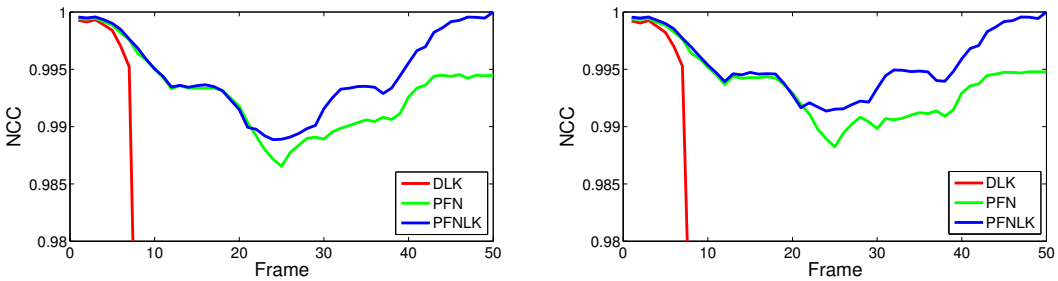


Figure 3.7: Comparison of performance for a FB sequence with camera motion. The first frame (frame 0) and the last frame (frame 50) are identical, so if a ROI is perfectly tracked, it should return to the initial location in the first frame: (a) Frame 0; (b) The DLK frame 50; (c) The PFN frame 50; (d) The PFNLK frame 50.

used to evaluate the tracking performance quantitatively. Higher NCC value is a surrogate measure for better tracking performance as it shows close image alignment. We computed the NCC between the template ROI and the tracked ROI in Figure 3.8a and computed it again after the SCV illumination mapping step shown on Figure 3.8b. The similarity between Figure 3.8a and Figure 3.8b means the illumination changes in this sequence are not large. As the figure indicates, the PFNLK method outperformed the PFN method which slowly drifted away during the backward tracking.



(a) **(b)**

Figure 3.8: The comparison of NCC and of tracked point with different tracking methods throughout the FB sequence: (a) NCC between the original template and tracked ROIs; (b) NCC computed after SCV illumination mapping step.

We chose another radical prostatectomy sequence[1] to evaluate the tracking performance in the presence of instrument occlusions. Occlusions are commonplace throughout the surgical

[1] http://www.surgicalvision.cs.ucl.ac.uk/data

procedure and present a significant challenge to tracking algorithms, especially if the instruments deform and manipulate the tissue of interest. This sequence consists of 600 frames and the tracked ROI is occluded by the surgical tools during certain time periods during the sequence.

We initialized each of the three methods with the same ROI and representative tracking results over the full sequence are shown in Figure 3.9. Since the DLK method encountered tracking failure quickly, we left it out of the discussion below. It is possible to observe that on frame 81 (the second column), the tracked ROI is occluded by the surgical tool, and for the following frames after the occlusion the PFN optimization of the mesh is trapped in local minima and never recovers back, as seen on frame 124 (the third column). By using the PFNLK method the tracked ROI recovered after frame 124 and continued tracking suggesting that the algorithm is more robust compared to the PFN method.

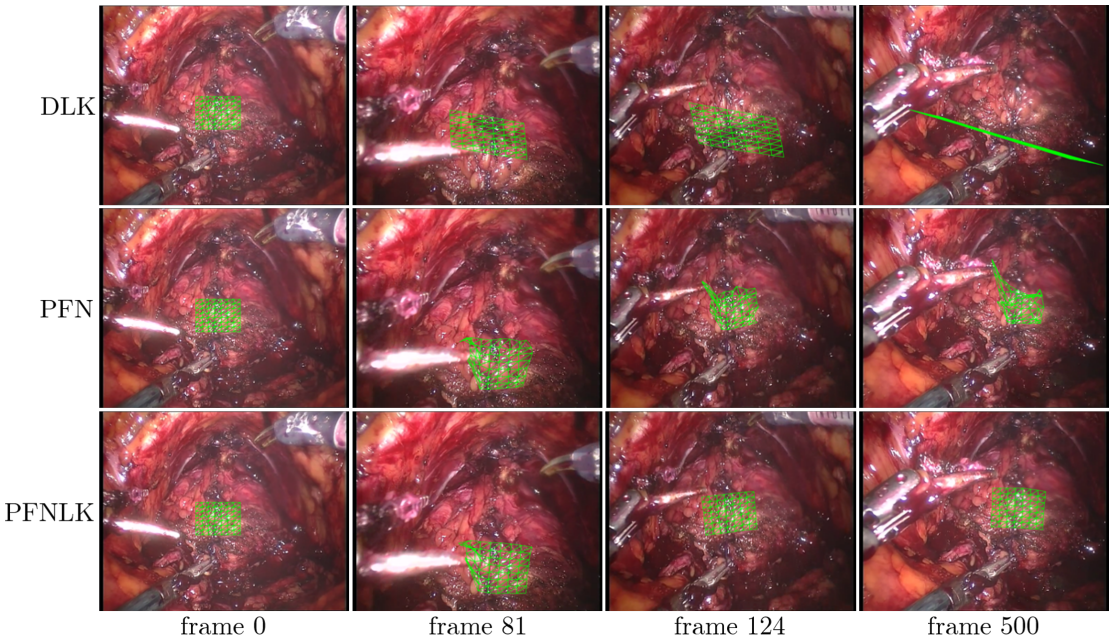


Figure 3.9: Comparison of performance for occlusion sequence: (Top row) intensity-based DLK method; (Middle row) feature-based PFN method; (Bottom row) our hybrid PFNLK method.

Following the same validation procedure, we computed the NCC between the template and the tracked ROIs for this sequence without and with the illumination mapping step. The results are shown in Figure 3.10. A sharp drop-off and recovery can be observed between frames 172 and 308 without the illumination mapping in Figure 3.10a, however, this is not reflected by visually inspecting the quality of the tracking results. In the original sequence the tracked areas were shifted to the very left side of the view during this interval, so the illumination condition changed greatly due to the camera motion. This inauthentic change reflects that the NCC metric cannot handle non-linear light changes very well. In Figure 3.10b, the input images are mapped to mimic the illumination condition of the template image using the SCV metric, and we can see that the inauthentic changes of the NCC have disappeared, also the NCC went up after frame 123 for the PFNLK method, which is accordance with the visual interpretation. This means that the NCC can be trusted as a surrogate measure under similar lighting conditions. We can

also infer from the evaluation that the SCV metric we used is necessary if there exist potential illumination variations.

Furthermore, we manually tracked one point within the ROI through the whole sequence, and use the position of the point in each frame as GT. In Figure 3.10c the tracked trajectories of different methods are illustrated with the GT, also the tracking errors are computed and shown in Figure 3.10d, we can see that the tracking error of the PFNLK is the lowest, which is consistent with the analysis above.

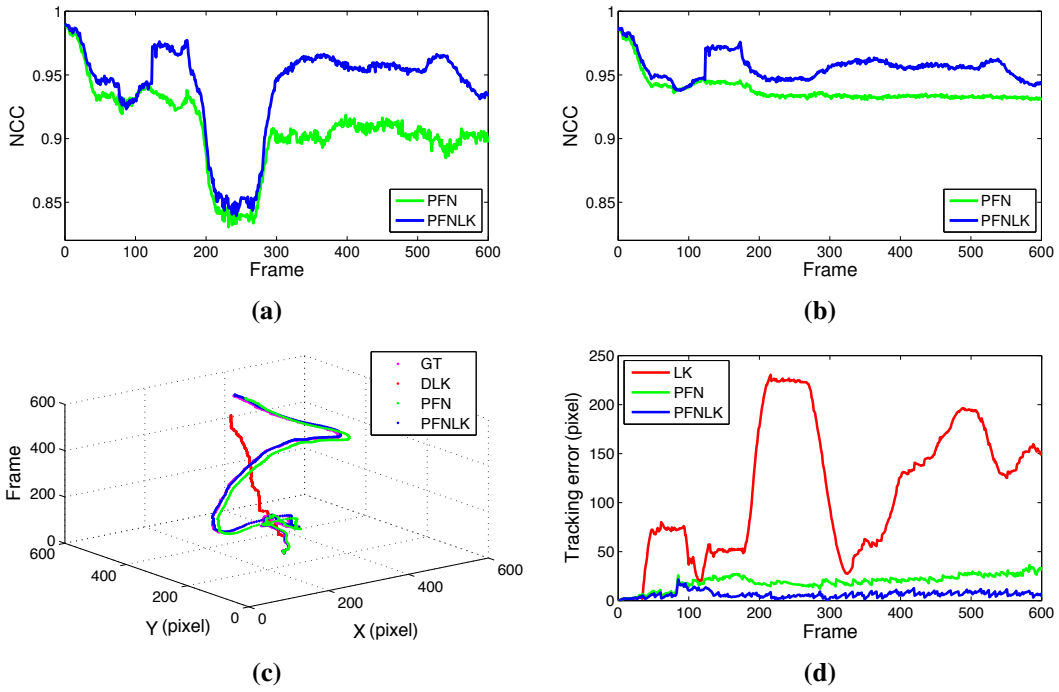


Figure 3.10: The comparison of the NCC and of tracked point with different tracking methods throughout the occlusion sequence: (a) The NCC between the original template and tracked ROIs; (b) The NCC computed after SCV illumination mapping step; (c) Trajectory of the tracked point; (d) Tracking error of the tracked point.

3.7.3 Experiments with Multispectral Data

To illustrate the importance of using the SCV rather than an illumination sensitive metric, we provide exemplar results of registering multispectral images. In sequential multispectral images where the image stack is acquired one wavelength at a time, some images can have very low signal strength due to the camera and light-tissue interaction characteristics such as absorption and scattering. Our multispectral image sequences from $\lambda = 480\ nm$ to $\sim 680\ nm$ is shown in Figure 3.11. It is not obvious from the figure, but the tissue under interrogation moves during acquisition, this causes misalignment of the multispectral stack and, for instance, renders spectral analysis to calculate oxygenation levels impossible [177].

We tracked the same ROI using the DLK method with and without the SCV illumination mapping step and the tracking result are shown in Figure 3.12. The histogram of the SCV images in the figure is equalized because the template image is too dark. As it is shown, the original DLK method in [167] loses track eventually (see top row); while our modified DLK method maps the input image I to the template image T in order to obtain the SCV image $\hat{I}$,

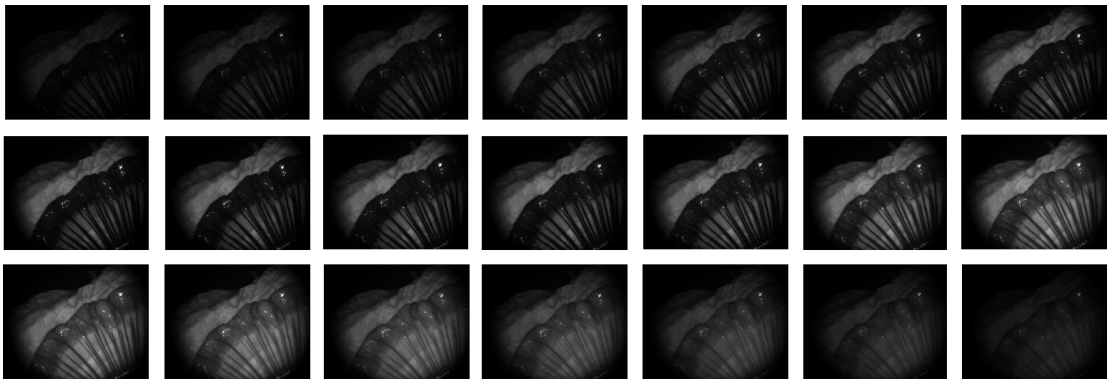


Figure 3.11: Multispectral images acquired one wavelength at a time from $\lambda = 480\ nm$ to $\sim 680\ nm$

after the SCV mapping step the images of different wavelengths are under similar illumination conditions (see middle row). The tracking result of our modified DLK method is more robust than the original DLK method (see bottom row).

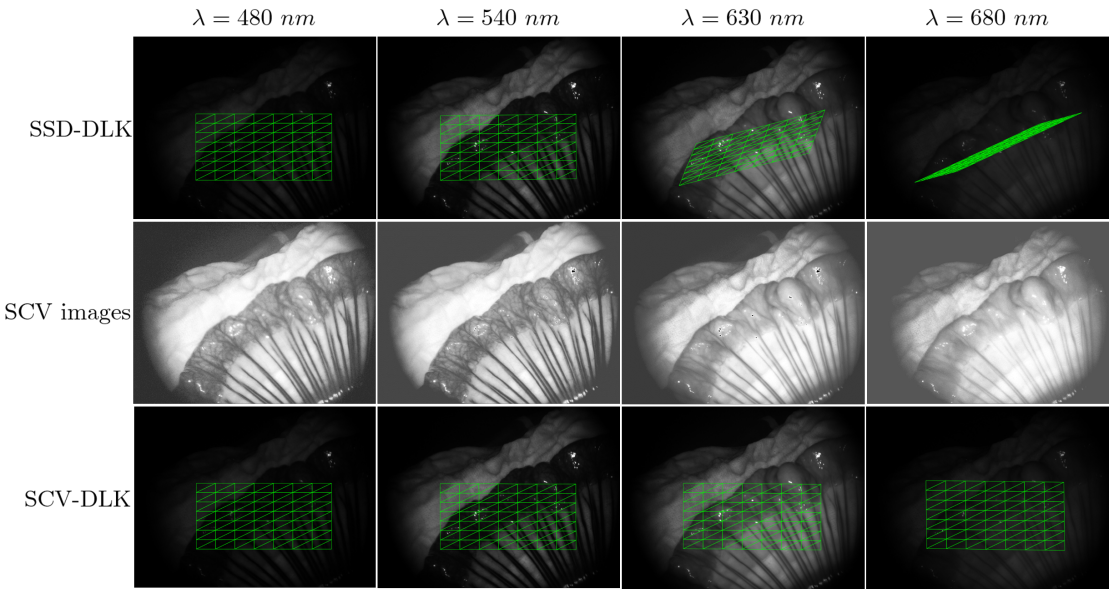


Figure 3.12: The alignment of multispectral images (wavelength $\lambda = 480 \sim 680\ nm$) without and with illumination compensation: (Top row) original DLK method using SSD metric; (Middle row) SCV images; (Bottom row) our modified DLK method using SCV metric.

Since the tissue motion is quite small in the sequence, to show our tracking effect more clearly we picked images of wavelength from $550 \sim 570\ nm$ with observable motion from the sequence and computed the absolute difference between the template and tracked ROI without and with misalignment correction. Due to the darkness, the result image is enhanced, and transformed to pseudo-colour image as illustrated in Figure 3.13. It shows that the misalignment decreases with our method, and the spectral data can be reconstructed after the motion compensation.

We also tested on other multispectral images, and showed the difference image result in Figure 3.14 and Figure 3.15. The misalignment of vessels on 3.15 is corrected using the SCV metric.

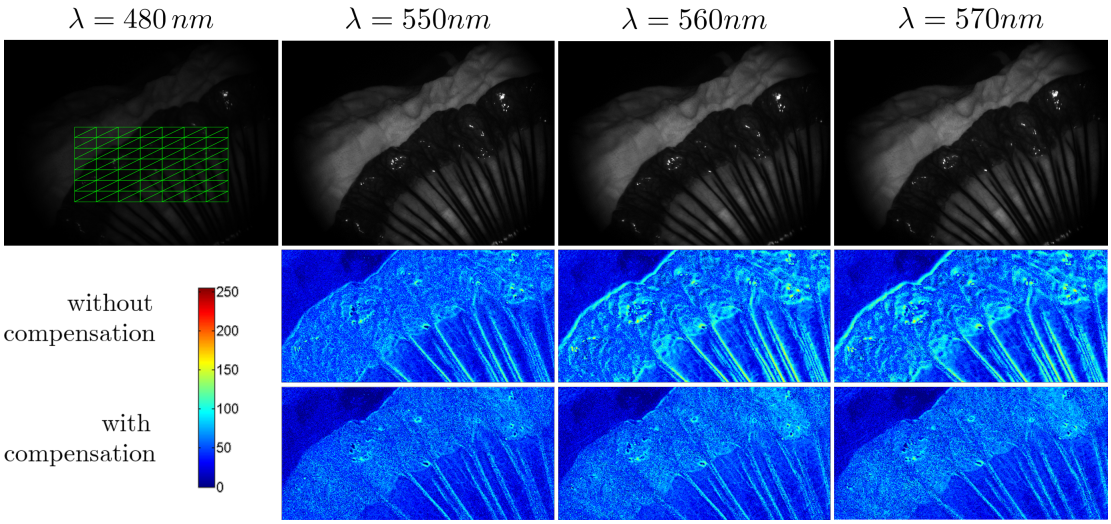


Figure 3.13: The original multispectral images and the difference images without and with using SCV metric: (Top row) the template frame ($\lambda = 480nm$) with the tracked ROI and several frames with observable motion; (Middle row) the difference ROI images without compensation; (Bottom row) the difference ROI images with compensation

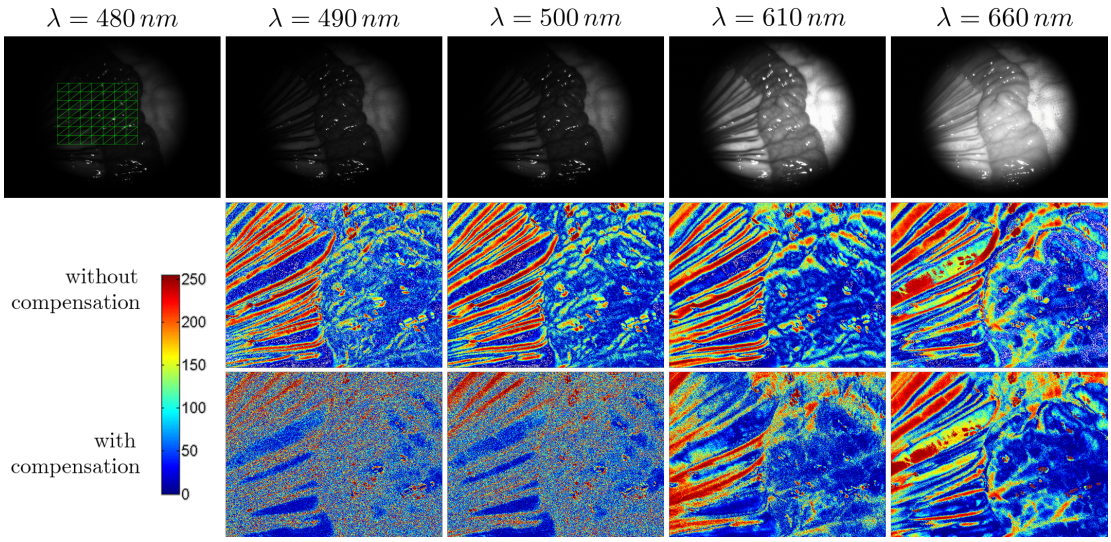


Figure 3.14: The original frames and the difference images of another multispectral image sequence.

During the computation of the probability distribution functions in Equation 3.11, noise will be added to the SCV images due to the impact of histogram binning [178]. If the chosen number of histogram bins is too low, the resulting SCV images will lose a lot of high frequency details; on the other hand, if the full dynamic range of the image is chosen this results in noise in the SCV images. The impact may be alleviated by using adapted histogram bins. In our experiments, $d_T = d_I = 256$.

3.7.4 Experiments with Tongue Data

Like the multispectral registration in Section 3.7.3, our algorithm can serve as a pre-processing module for various applications. MSI is a promising technique by providing haemoglobin concentration in tissue, which can be used in MIS to monitor organ viability or detect abnormal tissue [179, 180]. However, the MSI techniques are limited for real-time requirement due to

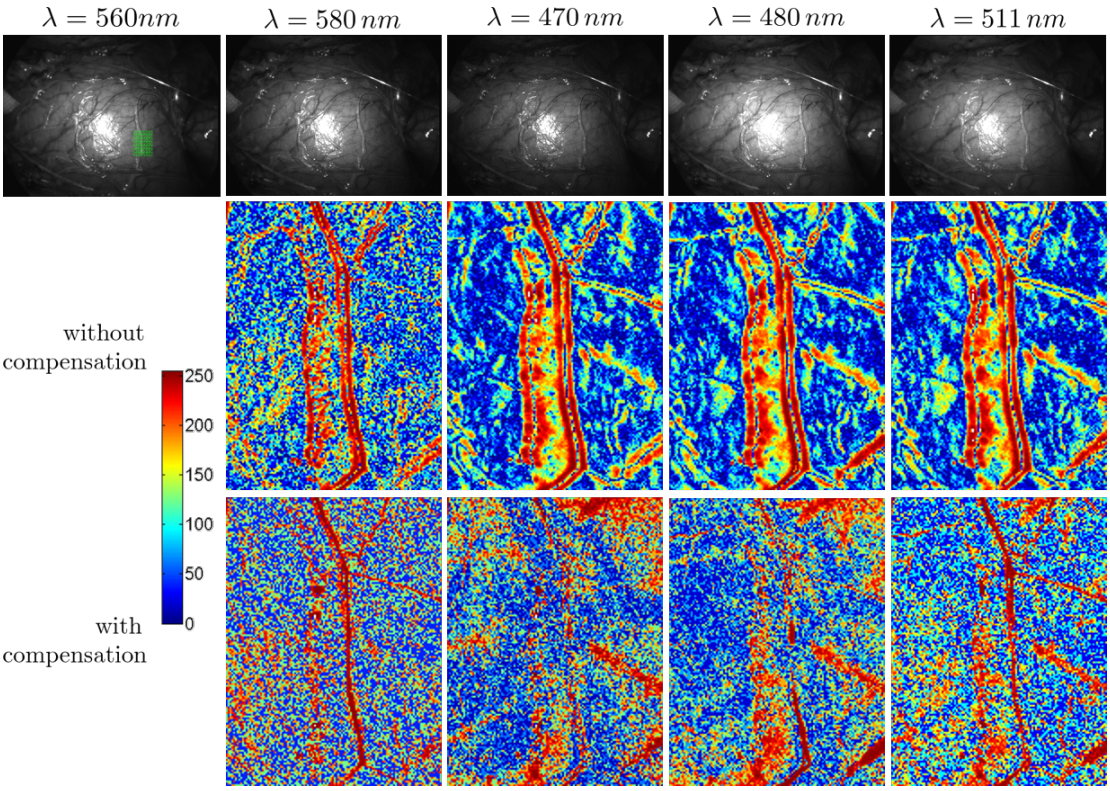


Figure 3.15: Vessel misalignment correction.

its capture rate or processing speed. Special hardware [181] has been designed to alleviate the problems but is impractical for MIS surgical environment. Therefore, vision-based methods which utilize the laparoscopic RGB video feed as a surrogate has attracted wide interest. An important and inevitable pre-processing step is to compensate the motion of monitored tissue.

Using a Da Vinci surgical robot, we recorded a video of the tongue base of an adult male using a da Vinci surgical robot's stereo laparoscope [182]. We showed some of frame examples from the sequence in Figure 3.16. Due to the tongue movement, dense tracking or registering are needed to remove the residual motion artefacts to obtain the spectral and temporal variation within a registered spatio-temporally signal. We selected one ROI on the first frame from the left camera and used this frame as the reference, the tissue patch is registered in both left and right camera over time separately. After the motion compensation, we compared the mean value of total haemoglobin (THb) from each camera view, and a strong similarity can be observed. By analysing the change in THb over time, the peak of the spectrum resonated with the heart rate of the subject during the data acquisition.

3.8 Discussion

In this chapter, we presented a hybrid tracking method for estimating the deformation of soft-tissue surfaces by using a constrained geometric model combining sparse feature tracking with a modified DLK method [167]. Our algorithm uses the SCV as the similarity metric to handle illumination variations, for example as seen in MSI, and facilitate tracking in very low light conditions where traditional approaches fail. The performance of our method on synthetic and *in vivo* datasets suggests that the hybrid approach improves the capability of correct

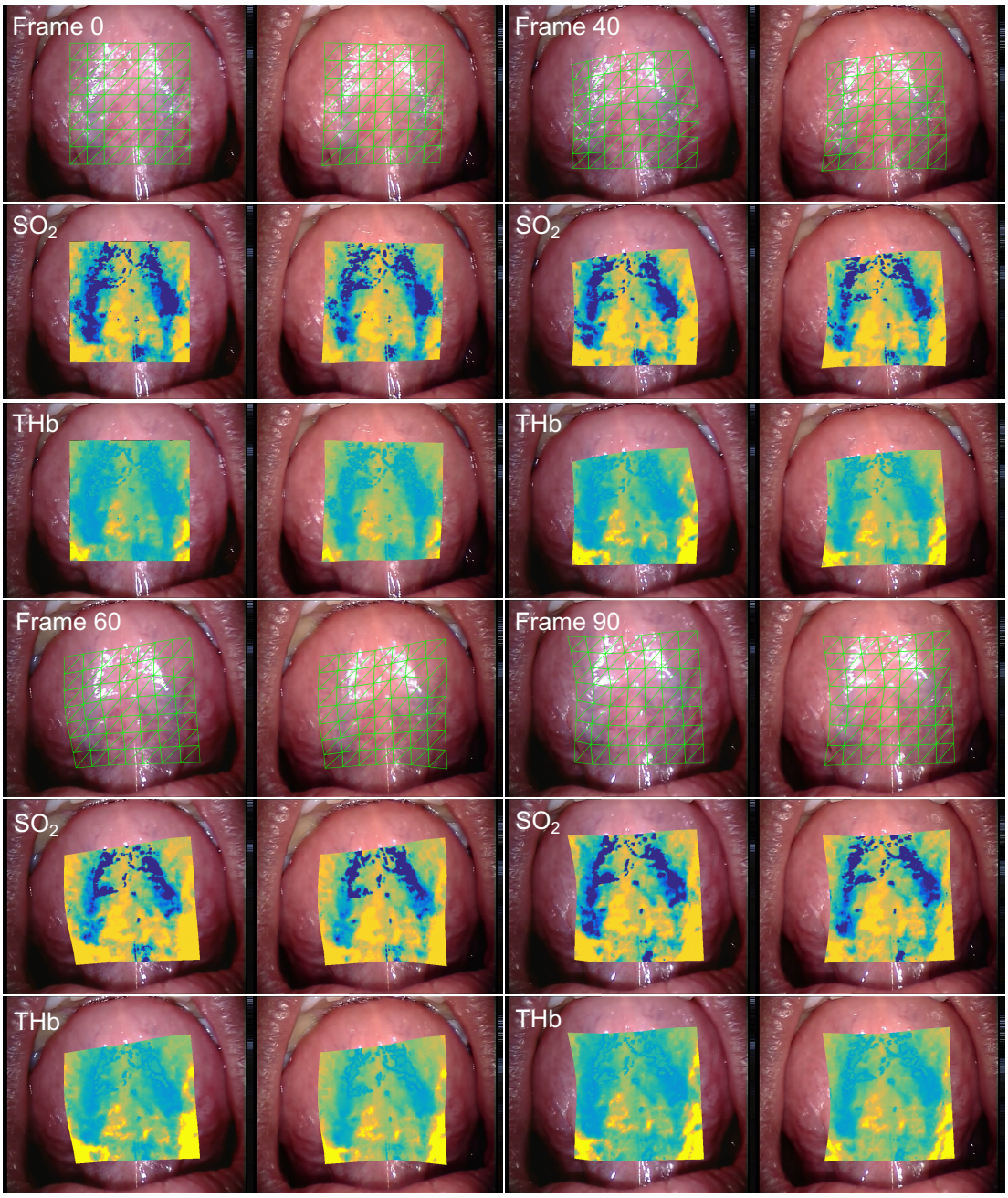


Figure 3.16: Tongue tissue motion is compensated by tracking over time in both left and right cameras separately, the registered outputs are used for oxygen saturation (SO_2) and total haemoglobin (THb) estimation: (The 1st and 4th row) tongue tissue patch registration result; (The 2nd and 5th row) SO_2 estimation overlaid on the laparoscopic RGB image; (The 3rd and 6th row) THb estimation overlaid on the laparoscopic RGB image.

convergence when tissue dynamics undergo large intra-frame displacements or significant illumination change occurs. The feature tracking component of the proposed algorithm is very fast when using simple features and the mesh optimization using only the feature energy is computationally efficient allowing real-time application. In our work, we use the L^2 norm in our energy functions for the optimization (Equation 3.2 - 3.6), which is known for penalizing large discontinuities and favouring smooth solutions. Inspired by the development of Total Variation [183], one of the possible future works is to explore more loss functions, such as the

TV-L^1 model, in which the regularization term is replaced by the Total Variation norm, and the residual term is replaced by L^1 norm. The Total Variation regularization does not suffer from losing contrast, and preserves geometry, while the L^1 norm is more robust and insensitive towards outliers. A GPU implementation of the improved intensity-based DLK component using SCV would accelerate the current algorithm and we believe can also be developed to work at image-acquisition frame rates. Extending the presented algorithm to stereo images is also potentially interesting future work, however, the regularization that we currently employ requires modification to appropriately handle deformations in 3D space.

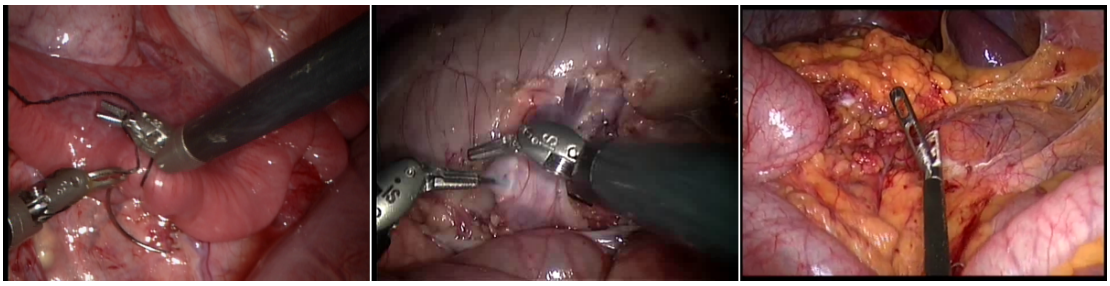


Figure 3.17: Tissue surface tracking is commonly affected by the intrusion of the surgical instruments in restricted surgical procedures.

Notably, our method is a general non-rigid tracking framework, and can be applied to other tracking tasks. The tissue surface is represented by deformable geometric model, but it is not designed for soft tissue surfaces. Specifically, the warp function $\mathbf{W}(\mathbf{p};\mathbf{S})$ used for transforming pixels from the input image into the template image coordinate is piecewise affine. This means the number of vertices of the deformable mesh has to be large enough to capture the deformation of the surface effectively. Besides, the stiffness of the model is controlled by the regularization parameter λ. The higher the regularization parameter is, the more bending or deformations are smoothed, and vice versa. Since it is set empirically, it does not incorporate any specific biomechanical tissue material properties. This inevitably limits the employment in real surgical procedures. Also as shown in Figure 3.9 and Figure 3.17, restricted *in vivo* surgical environment makes tissue surface tracking more challenging, one of the most common interferences is the constant intrusion or interaction of surgical instruments. In the following chapters, we are going to explore related surgical vision topics such as instrument tracking and pose estimation.

Chapter 4

Keypoint Based Surgical Instrument Tracking

4.1 Introduction

Detection and tracking of surgical instruments can provide an important information component of computer assisted surgery (CAS) for MIS [28]. Control systems which can supply automated visual servoing [184], soft motion constraints [185] and tactile feedback [186] are reliant on knowing positional information about both the shaft and the tip of the articulated instrument. Hardware based solutions such as optical tracking systems using fiducial markers [187] require modification to the instrument design posing ergonomic challenges and additionally suffer from robustness issues due to line-of-sight requirements. Direct use of robotic joint encoders and forward kinematics to track instruments is possible in robot-assisted interventions, however, tendon driven systems, such as the da Vinci® (Intuitive Surgical Inc., CA) introduces errors in the position information which usually requires correction that can be achieved through visual methods [188, 184]. Entirely image based solutions [137, 189, 190] directly estimate the instrument pose in the reference frame of the observing camera. This avoids complex calibration routines and can be implemented entirely through software which allows them to be applied retrospectively and without modification to the instruments or the surgical workflow.

Early image-based methods predominantly estimated the instrument pose in 2D by estimating image based translation parameters, scale and in-plane rotation without explicitly modelling the 3D shape of the instrument. These have been based around low-level image processing [191] which accumulate hand-crafted visual features and more complex learned discriminative models [192, 190] which track an instrument by performing detection independently on each frame. Such methods are typically fast and robust, handling complex and fast motion as well as recovery when the instrument is occluded by the field of view of the camera or smoke and tissue as they perform a global or semi-global search of the entire image for the tracked instrument. Fewer methods have attempted to estimate the 3D pose of the instruments directly from image data. This typically is a much more complex problem as it involves estimating three additional DOF from very weak small baseline stereo or monocular cues. However, it provides additional benefits over 2D methods as it allows reasoning about instrument-instrument occlusions and interactions with tissue surfaces. Most of these methods focus on the alignment of a 3D model with a probabilistic classification of the image [135, 136, 134] which allows the fusion of geometric constraints with image data without an offline learning phase. A significant challenge with 3D tracking methods is that they commonly fail when the instrument motion is

fast or complex. As shown in Figure 4.1, we illustrated several typical failure cases where pure 3D tracking drifts away and requires a manual reset. Generally, most tracking by model fitting 3D tracking methods restrict the parameter search to local regions close the estimated parameters from the previous frame, it locally searches the most likely parameters where the model is the most similar to the target appearance, so when the instrument is occluded, whether by other targets or out-of-view, the algorithm would fail. Also, the tracking error usually propagate and accumulate over long term tracking, which eventually lead to target drift.

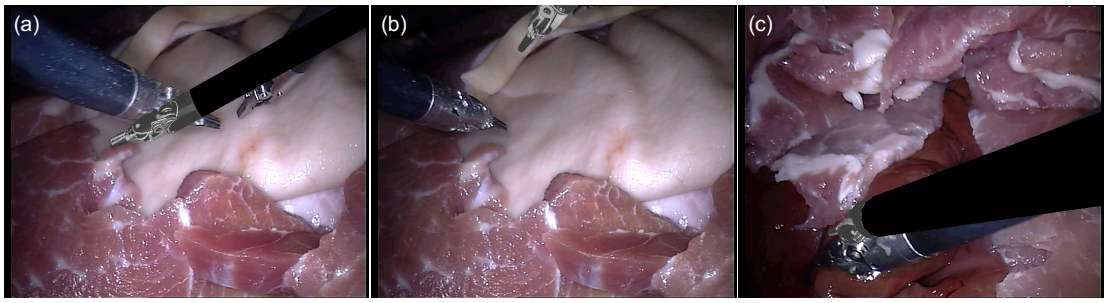


Figure 4.1: Challenges for 3D tracking methods: (a) Illumination by other instruments; (b) Out-of-view; (c) Long-term tracking drift.

4.2 Tracking Pipeline

Our method assumes that we have the 3D pose of the instrument in the first frame which we use to initialize a 2D bounding box (u', v', w, h) (see Figure 4.2) around the instrument head where (u', v') is the pixel coordinates of top left corner of the bounding box which has width w and height h. We define the 2D detection problem as the estimation of the parameters $\lambda_{2D} = (u, v, \theta, s)$ and the 3D estimation problem as the estimation of the parameters $\lambda_{3D} = (x, y, z, \phi, \psi, \hat{\theta})$, where (u, v) are the pixel coordinates of the centre of the instrument head, θ is the pitch/in-plane rotation of the instrument shaft around the optical axis, and s is the scale of the tracked target. (x, y, z) are the 3D translation coordinate in metric units from the camera coordinate system origin to the instrument coordinate system origin, $\phi, \psi, \hat{\theta}$ are the x, y, z rotations of the instrument in 3D respectively. For each new input frame, we detect the instrument, estimating the 2D parameters λ_{2D} using our new tracker. Using these parameters we then initialize a previously developed, open-source 3D tracker [137] which then converges using gradient descent to estimate the full 3D parameter vector λ_{3D}.

4.3 Generalized Hough Transform for 2D detection

To estimate λ_{2D}, we implement a keypoint-based tracker which relies on a GHT [193] and a global histogram segmentation model. The GHT extends the well-known Hough Transform to detect arbitrary shapes as maxima in a parameter space by describing shapes as collections of spatial features in a local coordinate system. As shown in Figure 4.3, a shape is defined by its boundary points and a reference point (u, v). For each of the boundary point (x, y) , the displacement vector is computed in the form of radial distance r and the angle α with regards to the reference point, and they are stored in a table indexed by the gradient orientation β, which is referred as the R-Table. Note that for each orientation, there may be multiple values of (r, α).

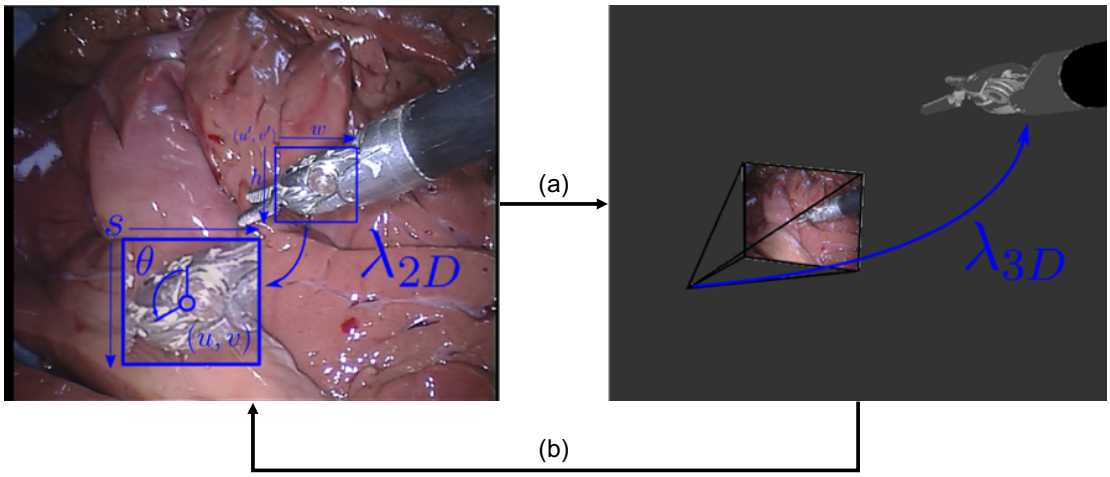
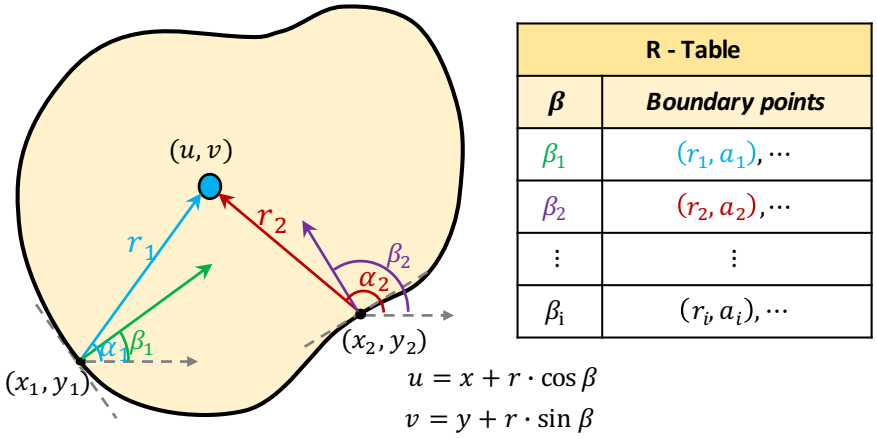


Figure 4.2: The left image shows the 2D detection and estimation of the parameters λ_{2D} which are then are used (a) to initialize the 3D parameters λ_{3D}. After the 3D pose is estimated, a new frame is loaded (b) and 2D detection begins again.

Assuming the scale and orientation are fixed, for each boundary point in the test image, the properties of the point is looked up in the R-Table according to its gradient orientation, and the correspoinding possible reference points are retrieved and accumulated, which is referred as "voting". Finally, the position with the maximum votes are considered as the reference point in the test image. In our application, the target is defined by keypoints and a reference point.



R - Table	
β	*Boundary points*
β_1	$(r_1, a_1), \cdots$
β_2	$(r_2, a_2), \cdots$
⋮	⋮
β_i	$(r_i, a_i), \cdots$

Figure 4.3: Shape detection using Generalized Hough Transform

Given an example image template containing the object of interest, a reference point which serves as the origin of the local coordinates is computed, usually as the centre of the template window. Then, for keypoint based features (e.g. SIFT [194]) in the template image, the feature orientation and the relative displacement and orientation to the reference point are computed and stored in a database known as an R-Table, which fully defines the target object. To perform detection with the GHT, keypoints in a new image are computed and matched to the stored keypoints in the R-table. Each matched keypoint then 'votes' for the origin of the coordinate system and the centre is chosen as the reference point with the most votes.

4.4 Model Initialization

Given a sequence of m frames $\{I_t\}_{t=1}^{m}$ and the 2D bounding box (u', v', w, h) on the template frame I_1, we detect the parameters $\lambda_{2D} = (u, v, \theta, s)$ on every input frame. The object model M is represented by a set of keypoints

$$M = \{(\mathbf{f}_{i,t=1}, d_i, v_{i,t=1})\}_{i=1}^{n} \tag{4.1}$$

where $\mathbf{f}_{i,t=1}$ denotes the descriptor of the ith keypoint on the model, d_i represents the distance between keypoint $\mathbf{f}_i$ and the centre of the instrument head (u, v). $v_{i,t} \in \{0, 1\}$ is the voting state of the ith keypoint at frame t: 0 for negative, and 1 for positive. It is positive if the corresponding keypoint has contributed for the voting of the detected centre, otherwise is negative. The voting states for all keypoints are initialized as positive for the template frame I_1

$$v_{i,t=1} = 1 \quad \forall i \in [1, n] \tag{4.2}$$

For each input frame I_t with $t > 1$, the keypoints in the model are matched. We gather the descriptors of the matched corresponding keypoints as the vote set F_V.

$$F_V = \{(\mathbf{f}_{i,t}, w_{i,t})\} \quad \forall i \in [1, n] \tag{4.3}$$

where $w_{i,t}$ is the voting weight for the corresponding ith matched keypoint, which is defined based on the segmentation model introduced in section 4.5.

4.5 Histogram-based Segmentation Model

To adapt object model accounting for appearance changes, we are inspired by the work of [195, 196] and we implemented a global probabilistic model based on colour histogram by using a recursive Bayesian estimation to better discriminate foreground and background.

Recursive bayesian estimation, which is known as Bayes filter, is a general probabilistic method for estimating the probability density function recursively over-time using the measurements and the Hidden Markov model (HMM). In Figure 4.4, the true states $\mathbf{x}$ are assumed to be Markov process, which are not directly visible, and the measurements $\mathbf{z}$ are the observed states of a HMM. The graph presents a Bayesian Network of a HMM. Based on the Markov

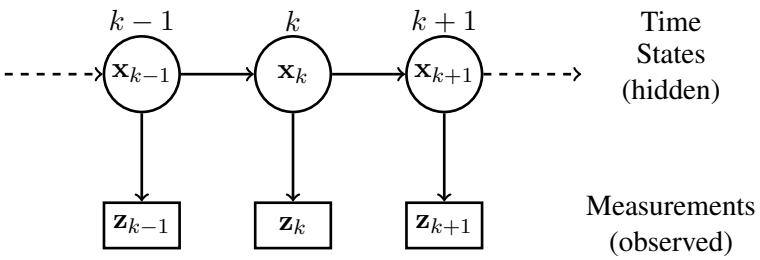


Figure 4.4: A Bayesian Network of a Hidden Markov model.

assumption, the probability of the k-th timestep state given the previous timestep $(k-1)$-th one

is conditionally independent of the other earlier states.

$$p(\mathbf{x}_k|\mathbf{x}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{4.4}$$

Similarly, the measurements at the k-th timestep is also only dependent upon the current state, which means it is conditionally independent of all the other previous states given the current state.

$$p(\mathbf{z}_k|\mathbf{x}_{1:k}) = p(\mathbf{z}_k|\mathbf{x}_k) \tag{4.5}$$

When we estimate the state $\mathbf{x}$, what we are interested is the probability distribution of the current state conditioned on the measurements up to the current timestep $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, which can be achieved by Bayesian marginalisation.

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1} \tag{4.6}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \tag{4.7}$$

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k \tag{4.8}$$

In practice, when computing $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, the denominator $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ usually ignored since it is constant relative to $\mathbf{x}$, and the numerator is calculated and normalized.

In our application, the measurements $\mathbf{z}$ are intensities of image pixels y, and the state $\mathbf{x}$ is the class (foreground or background). The probabilities of a pixel y belonging to the foreground or background is estimated recursively over-time based on Equations 4.6 to 4.8.

$$p(c_t = 1|y_{1:t}) = Z^{-1}p(y_t|c_t = 1)\sum_{c_{t-1}} p(c_t|c_{t-1})p(c_{t-1}|y_{1:t-1}) \tag{4.9}$$

where c_t is the class of the pixel at frame t: 0 for background, and 1 for foreground, $y_{1:t}$ be the pixel's colour from frame 1 to t, and Z is a normalization constant to keep the probabilities sum to 1. The classification probability of a pixel at frame t is based on the previous posterior $p(c_{t-1}|y_{1:t-1})$, the colour distribution $p(y_t|c_t)$ and the transition model $p(c_t|c_{t-1})$. The colour distribution $p(y_t|c_t)$ is built with HSV colour histograms with 12×12 bins for H and S channels and 8 separate bins for V channel (including a separate quantisation for pixels with low value). We omit the background probability $p(c_t = 0|y_{1:t})$ here since it is similar to Equation 4.9. The transition probabilities for foreground and background $p(c_t|c_{t-1})$ where $c \in \{0, 1\}$ are empirical choices as in [196], which are not very sensitive.

$$p(c_t = 1|c_{t-1} = 1) = 0.6 \qquad p(c_t = 1|c_{t-1} = 0) = 0.4 \tag{4.10}$$

$$p(c_t = 0|c_{t-1} = 0) = 0.6 \qquad p(c_t = 0|c_{t-1} = 1) = 0.4 \tag{4.11}$$

The bounding box is usually slightly larger than the object in order to include more boundary keypoints, which nevertheless introduces more background pixels (shown as the red bounding box in Figure 4.5 (a). Unlike in [196], the foreground colour distribution $P(y_t|c_t = 1)$ is

initialized directly using pixels from the bounding box, we use a different strategy. We assume that the positive keypoints are most likely located on the object, so we collect all the positive keypoint into F_{Pos}

$$F_{Pos} = \{\mathbf{f}_{i,t}\} \quad \text{if } v_{i,t} = 1 \quad \forall \mathbf{f}_{i,t} \in F_V \tag{4.12}$$

The foreground colour distribution is then initialized from the image region inside the convex hull of all the positive keypoints $CH(F_{Pos})$, which contains less background pixels (shown as the green convex hull in Figure 4.5 (a)). The background colour distribution is initialized from the image region surrounding the detected object bounding box with some margin (10 pixels) in between, which is shown as the blue region in Figure 4.5 (a). For the following frames, the colour distributions are adapted in the same way as the initialization.

$$p(y_t|c_t = 1) = \delta p(y|y \in CH(F_{Pos})) + (1 - \delta)p(y_{t-1}|c_{t-1} = 1) \tag{4.13}$$

where $\delta = 0.1$ is the model update factor.

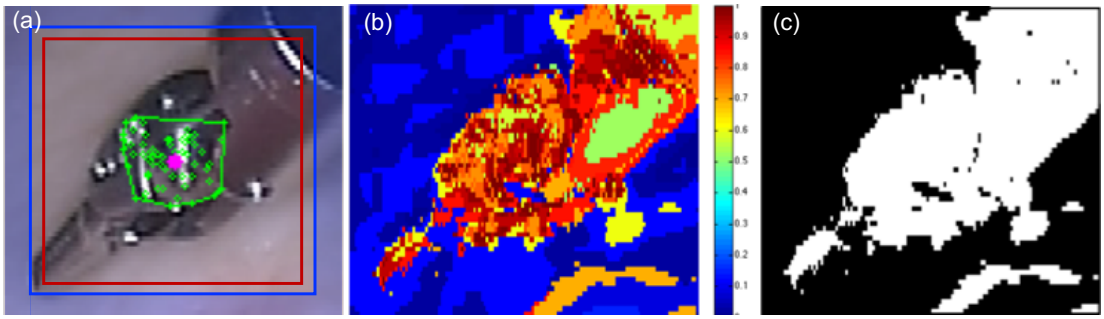


Figure 4.5: Segmentation model initialization and update strategy: (a) instead of using image region inside the bounding box (red), image region inside the convex hull (green polygon) of the positive keypoints (green circle) is used to initialize and update the foreground histogram. And background histogram is then initialized using pixels from the pixels outside of the surrounding bounding box (blue region). Filled circle with magenta colour indicates the reference centre; (b) foreground probability colourmap illustration, in which blue colour indicates low probability while red colour indicates higher probability; (c) foreground / background classification binary map based on the probability model.

The voting weight of a keypoint is defined as the mean foreground probability of the image patch surrounding the keypoint

$$w_{i,t} = p(c_t = 1|\mathbf{f}_{i,t}) \tag{4.14}$$

During the voting process, we set the weight threshold $w_{thres} = 0.5$, only keypoints with higher weight $(w_{i,t} > w_{thres})$ participate in the voting process, and the weighted votes accumulated based on the segmentation model. In regards to the voting, we developed a rotation-invariant voting scheme in section 4.6.

4.6 Rotation-invariant Hough Voting Scheme

When the object undergoes scale change or in-plane rotation (when the object rotates in the image plane), the voting also needs to rotate and scale in order to locate the object centre. Scale and rotation information can be obtained from most feature detectors, but since it is usually not reliable enough, in [1], the authors analysed the pairwise Euclidean distance and angular

change between keypoints with respect to their initial constellation and use the median as the scale or in-plane rotation estimate of the object. We illustrated their voting scheme and ours in Figure 4.6: Keypoints on the model and on the input frame are matched in Figure 4.6 (a1-a2), then in the input frame, median pairwise angular change between keypoints is computed by comparing with the initial constellation in Figure 4.6 (b1), and correspondent keypoints rotate votes based on the median angular change θ' in Figure 4.6 (b2). It displays the ideal situation for rotation estimation, but when the percentage of outliers is high, votes will probably miss shoot the centre based on unreliable rotation estimation. We develop a rotation-invariant voting strategy shown in Figure 4.6 (c1-c2). For each keypoint, instead of voting for only one direction, it votes for a circle. In this way, our vote scheme does not rely on any pre-estimation of rotation, the maximum vote still accumulated at the centre without any potential error induced by the pre-voting rotation estimation. In order to improve the over shooting or fall short situation for scale estimation or out-of-plane rotation (when the object rotates out of the image plane), we make it more robust by voting for a ring circle in Figure 4.6 (d1-d2). The thickness ratio r_d is set to be $[0.95, 1.05]$. The initial scale $s_{t=1}$ is set to be 1.0, the radius of the voting circle $d_{i,t}$ is based on the scale of the previous frame s_{t-1} and the distance of the keypoint to the reference centre of the model d_i

$$d_{i,t} = r_d * d_i * s_{t-1} \tag{4.15}$$

After voting, the scale s_t and rotation θ_t are estimated based on the scale change and angle change of all the positive keypoints.

4.7 Model Adaptation

One of the challenges for 2D visual tracking is how and when to adapt the object model to cope with appearance changes due to deformation, illumination variations, etc. In endoscopic images, when the object centre is out-of-view or out-of-plane, instead of updating the model, we have to reset the detector to re-detect the object. To achieve this, we define the following updating strategy. Whenever the voted centre is out of the convex hull of the positive keypoint set F_{Pos}, we evaluate all the keypoints inside the bounding box B_t around the detected centre based on the segmentation model. If the weight w_t^C of the keypoint candidate $\mathbf{f}_t^C$ is higher than the weight threshold w_{thres}, it is considered as a potential keypoint and is included in the keypoint candidate set F_{candi}, otherwise it will be discarded.

$$F_{candi} = \{\mathbf{f}_t^C\} \quad \text{if } w_t^C > w_{thres} \quad \forall \mathbf{f}_t^C \in B_t \tag{4.16}$$

Then, we analyse the distribution of the keypoint candidates with regards to the object centre: (i) if the centre (u, v) is inside the convex hull of the candidates F_{candi} and the number of candidates is higher than certain threshold, we add the new candidates M_{candi} into the model and remove negative features, then use the updated model to continue tracking; (ii) If the centre is outside of the convex hull, it indicates the object is most likely out of image or is under out-of-plane rotation, so we switch the detector into reset mode: If the object is matched, the detector will be switched back to normal mode.

We illustrate the tracking framework by showing some tracked frame examples from one

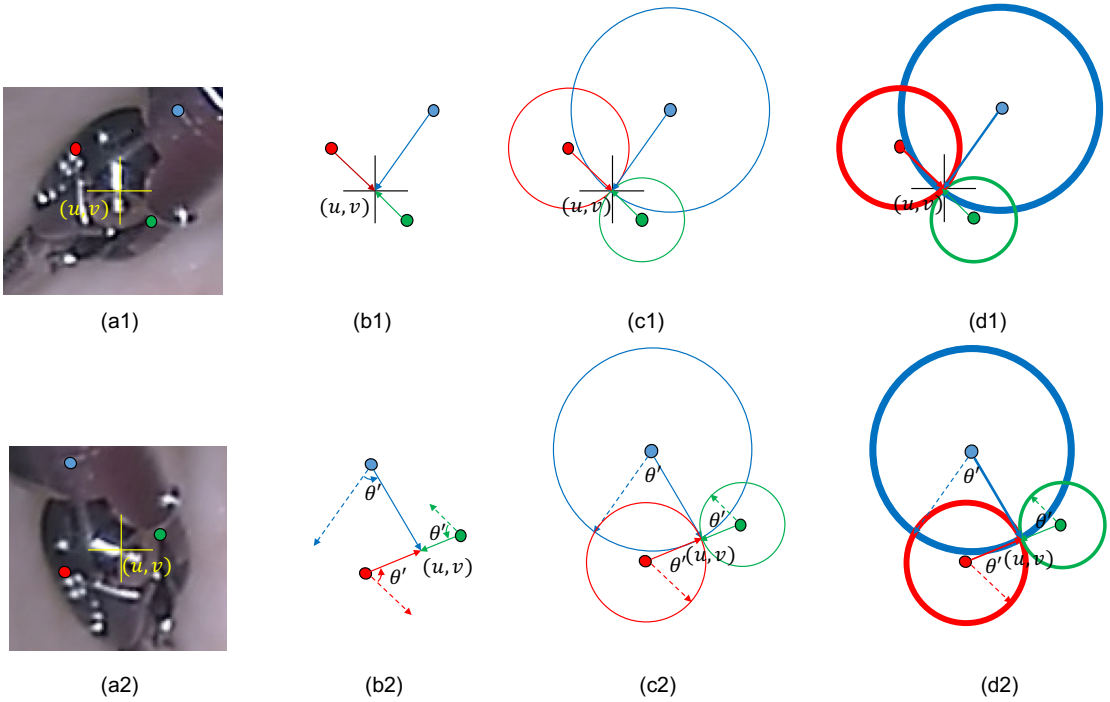


Figure 4.6: Voting scheme illustration: (a1) keypoints and reference centre on the model (shown in colour); (a2) keypoints and the tracked centre (u, v) on the input frame; in [1], keypoints vote for the reference centre (b1), in the input frame, the rotation θ is estimated by pairwise angular change and vote based on the rotation estimation in (b2); our rotation-invariant voting scheme votes not only for one direction but a circle (c1-c2), in order to improve robustness, keypoint votes for a ring circle, and the rotation θ and scale s are estimated after voting (d1-d2).

sequence in Figure 4.7. Each row represents one frame, with the left column displaying the voting map, the right column displaying the histogram-based foreground segmentation colormap, and the middle column showing the tracked result. In the sequence, keypoint features are extracted and matched for the right instrument, as we can see from the top row voting map, even with some outliers, correctly matched features vote and accumulate at the centre of the instrument. When the target is occluded by the other instrument in the middle row, votings accumulate at the left overlaying instrument, which shares similar appearance with the target, but it is obvious that the voting is not as concentrate as the top row, with increased erroneous keypoint matches. After the left instrument moves away, the target instrument got re-detected by the model correctly. Observing the foreground probability map, the segmentation model is less certain with the target pixels, since more and more pixels with the instrument colour entering the surrounding background region with the left instrument getting closer.

4.8 Combining 2D and 3D Tracking

We use an open-source 3D level set tracker [137] which is capable of recovering the full 3D pose of surgical instruments by aligning multiple level set segmentations with Random Forest pixel classifications and additionally uses optical flow tracking to local track features on the instrument body. We use the 2D pose λ_{2D} to initialize the 3D pose of this method in each frame, rather than using the tracking-by-initialization method of the original authors. The parameters (x, y, z) of λ_{3D} are initialized by ray casting (u, v) and using the z estimate from the first

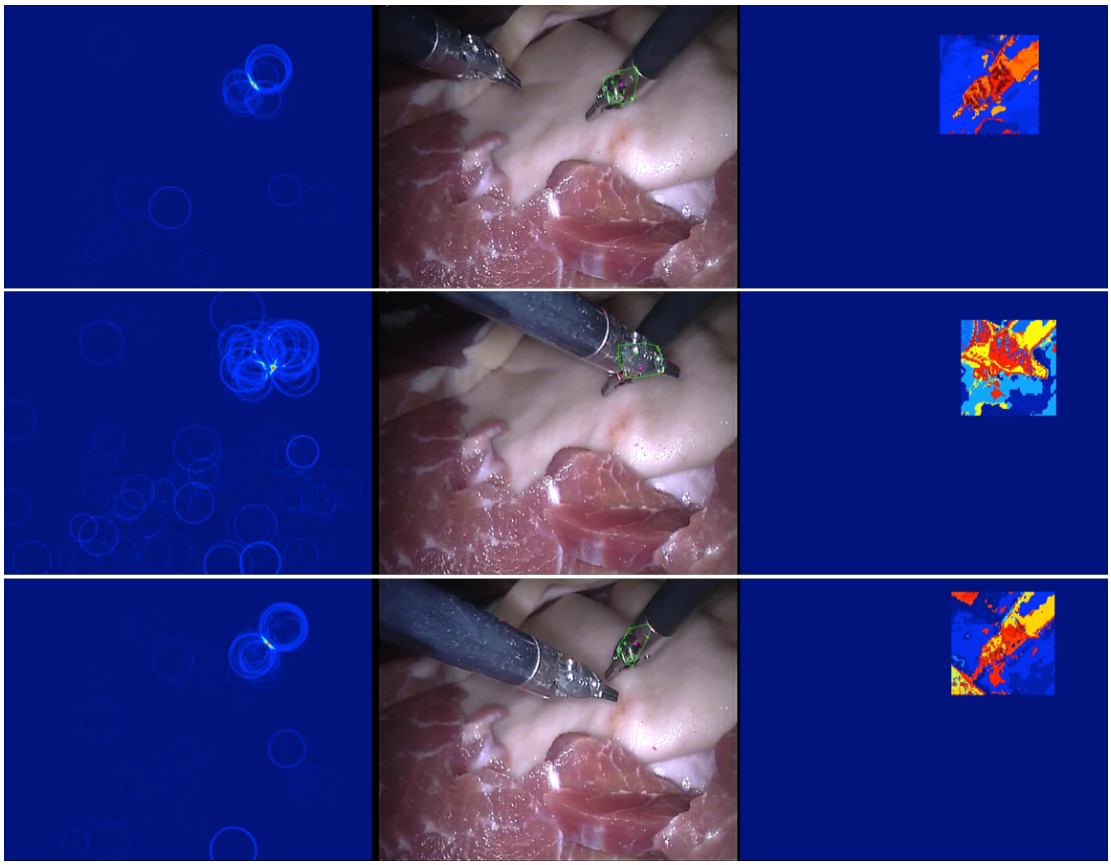


Figure 4.7: Tracking framework illustration. Each row shows one frame tracking example, left column shows the keypoint voting map, middle column shows the convex hull of the positive keypoints, and the right column shows the segmentation map in the search area.

frame, scaled by s. θ is used directly to initialize $\hat{\theta}$ and ϕ, ψ are retained from the previous frame. Effectively we only retain the parameters in the 3D tracker which cannot be estimated by the 2D tracker. Given an initial estimate we allow the 3D level set based tracker to converge to a solution through gradient descent.

4.9 Experiments and Results

In this section we present validation on both our novel 2D tracker (referred to as GHT) and our 2D-initialised-3D (referred to as 2D3D) tracking. In this section with refer to the 3D tracker without 2D initialization [137] as "3D only". Our quantitative validation is performed on new *ex vivo* data sets which we have made available online[1] (see Figure 4.8). We hope that by releasing data, we will encourage other researchers to test their methods against our data, an idea which was explored in the Endoscopic Vision Challenge at MICCAI 2015 which provided labelled segmentation and tracking data for laparoscopic and RMIS.

4.9.1 *Ex Vivo* Experiments

To evaluate the ability of our method to robustly track a surgical instrument through challenging sequences we constructed 4 datasets with porcine tissue samples. We have manually tagged the

[1] `www.surgicalvision.cs.ucl.ac.uk/benchmarking`

[1] To maintain notation consistency, the Shaft and End joint in our paper correspond respectively to End Shaft and Start Shaft joint in previous papers.

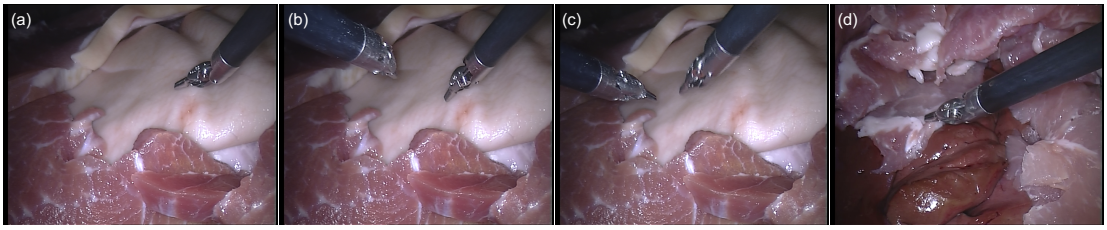


Figure 4.8: Example frames from our *ex vivo* sequences acquired using a da Vinci® (Intuitive Surgical Inc., CA) classic stereo laparoscope. The images show typical challenges in instrument tracking, such as instrument and tissue based occlusions and sequences where the instrument goes in and out-of-view repeatedly.

	Attributes	**Attribute / Frame Number**
Dataset I	Tissue occlusion	121 / 552
Dataset II	Instrument occlusion	130 / 909
Dataset III	Out-of-view	50 / 410
Dataset IV	Long term	- / 4483

Table 4.1: Attribute and percentage of frames which are tagged with the attribute for each dataset.

datasets with different attributes, including occlusion, out-of-view and long term. The percentage of frames which are tagged with the corresponding attribute are summarized in Table 4.1. Our *ex vivo* sequences are collected using a da Vinci® (Intuitive Surgical Inc., CA) robot where we obtained joint encoder data from a dVRK controller box [125]. Using forward kinematics we can compute the 3D transform for the instrument in the reference frame of the stereo camera using manual calibration to remove the offset between the robot and camera coordinate system. This can be projected into the image plane to obtain validation for both the 2D and 2D3D tracking. We compare our 2D tracking method with the-state-of-art CST tracker [6] and TLD tracker [3] using precision and box plots based on location error metric and area under curve (AUC) to analyse the performance. These metrics are widely used to evaluate tracking performance [81, 42]. Precision plots show the percentage of frames (y-axis) where the estimated position (u, v) is within a distance threshold (x-axis) compared with the GT. In the box plot, edges of the box are 25% and 75% percentiles, the whiskers extend to the most extreme data points not considered outliers, and the red markers are outliers plotted outside the box. We also summarise the numerical results for the 3D tracking in Table 4.2. In the table, mean translation errors for our 2D3D method and the 3D only tracking are shown for each of the *ex vivo* sequences.

Tracking Through Occlusions We evaluate on two different sequences with occlusions. The trajectories of the tracked centre and the precision plots for each sequence are shown in Figure 4.9 and Figure 4.11. In the figures, (a) shows the trajectories of the tracked centre for the three 2D methods, (b) shows the precision plot for three 2D methods, (c) shows the box plot for three 2D methods and (d-f) the 3D trajectory of the proposed 2D3D tracker compared with using the pure 3D tracker directly. We also showed some tracking frame examples in Figure 4.10 and Figure 4.12.

Dataset I evaluated the ability of the method to track instruments when they are occluded by tissue samples. The 3D tracker demonstrates similar performance with and without 2D

initialization in Dataset I, with slight improvement in the z-axis estimation during the occluded frames. In Figure 4.10, we show the original frames in the last row to display the instrument location more clearly. As we can see, after the instrument reappears from the tissue, there exists a offset between the pure 3D tracker result and the correct location. Our 2D3D tracker has more accurate tracking result.

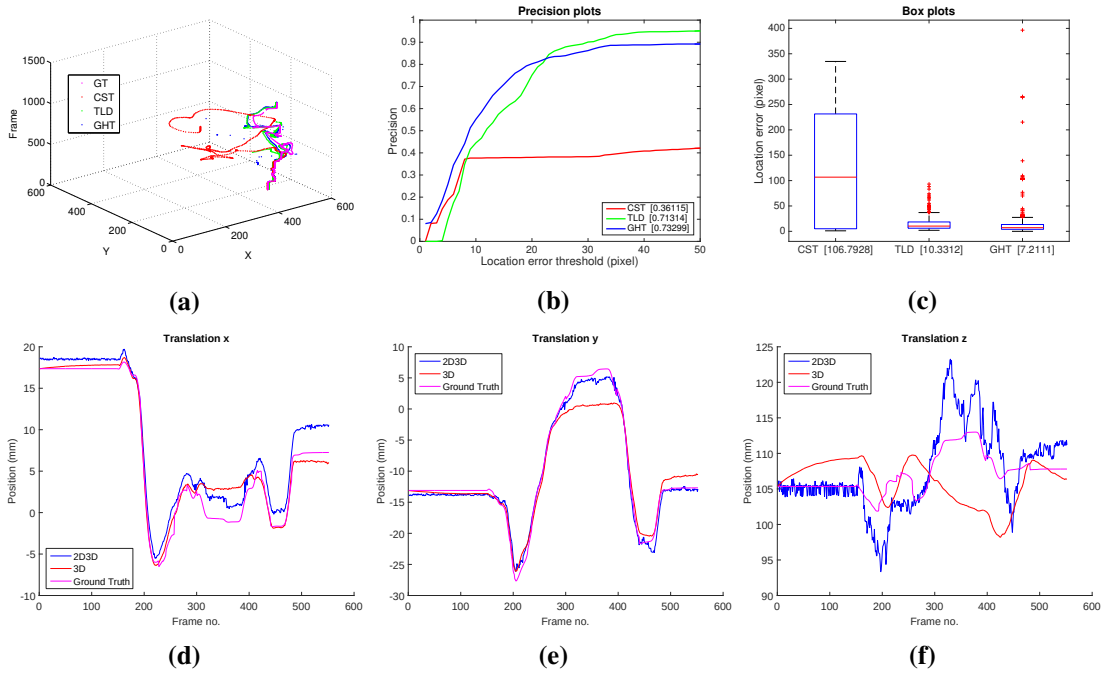


Figure 4.9: Performance comparison for Dataset I, which contains a tissue occlusion between frames 250-400.

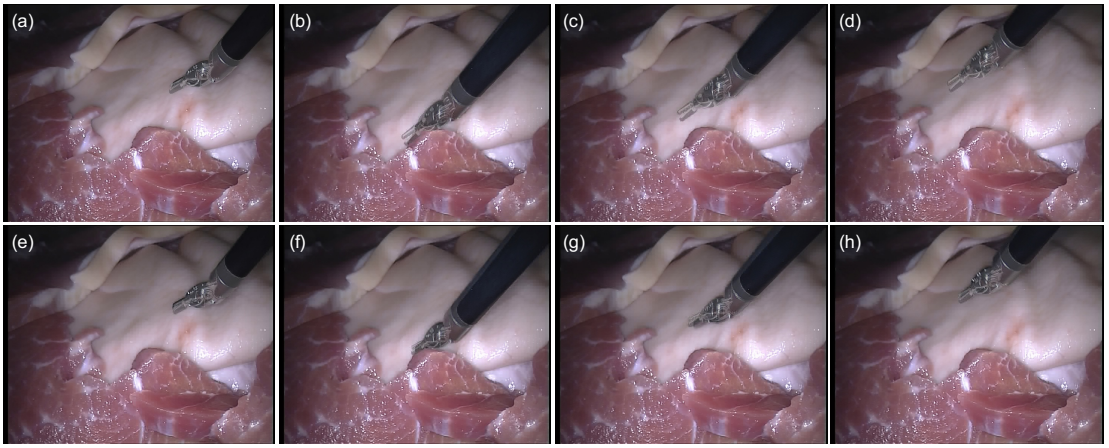


Figure 4.10: Frame examples of performance comparison between 2D3D tracking and pure 3D tracking under tissue occlusion. (a-d) Pure 3D tracker result; (e-h) 2D3D tracker result. To display the pose more clearly, the overlay is blended with the original frames.

Dataset II evaluates the ability of the method to track instruments when they are occluded by other instruments, effectively assessing our method's ability to avoid tracking association errors between the target instrument and additional instruments in the frame, even when they violate each other's image space. From Plot in Figure 4.11 and Figure 4.12, Dataset II clearly

demonstrates the improvement of our method as the 3D only tracker loses tracking at frame 380 and never recovers.

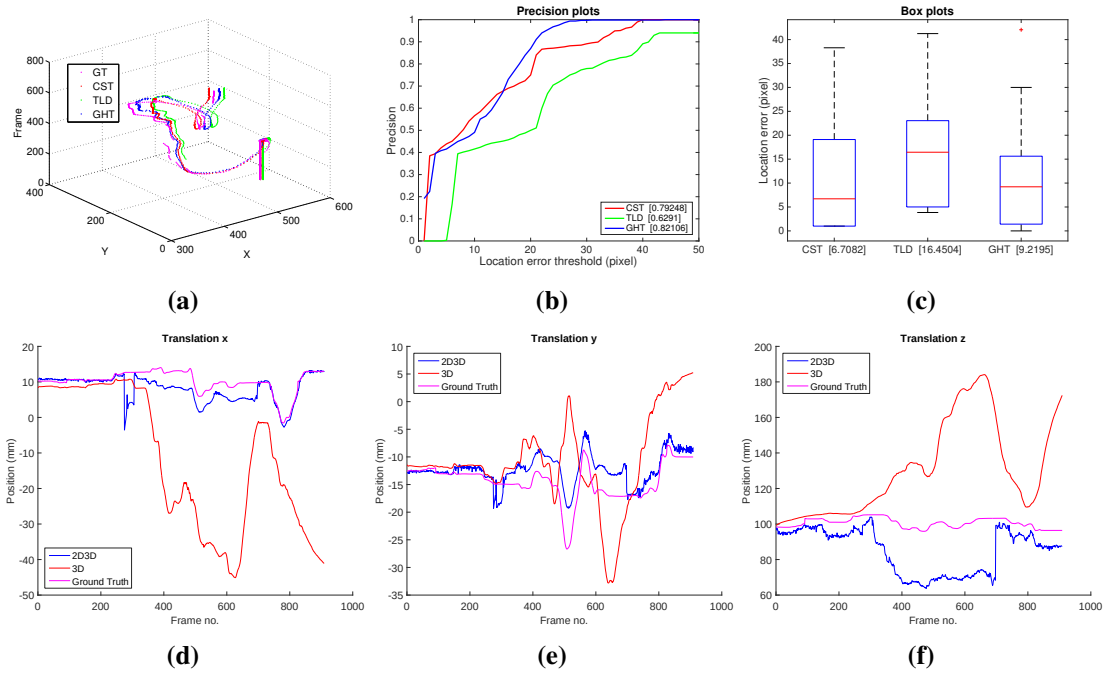


Figure 4.11: Performance comparison for Dataset II, which contains a instrument occlusion between frames 225-350.

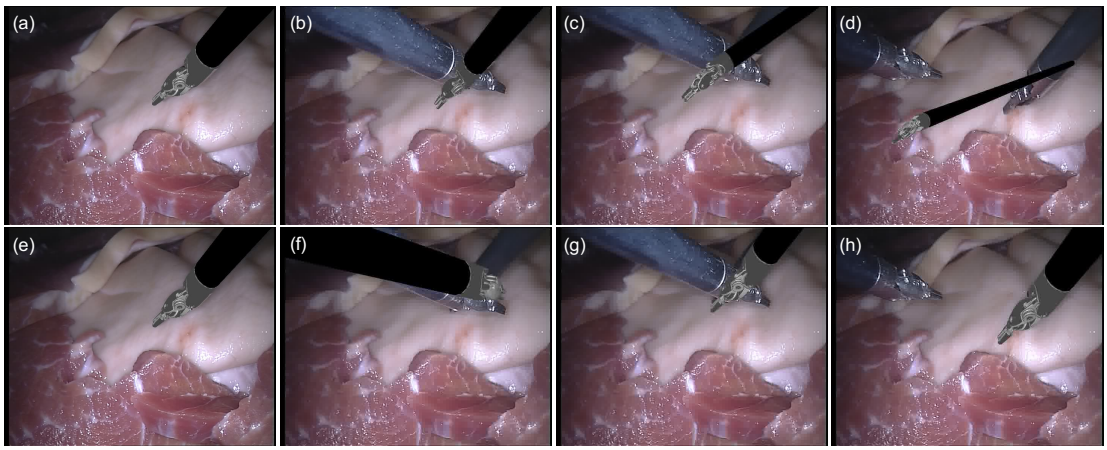


Figure 4.12: Frame examples of performance comparison between 2D3D tracking and pure 3D tracking under instrument occlusion. (a-d) Pure 3D tracker result; (e-h) 2D3D tracker result.

Tracking Through Out-of-view Dataset III evaluates the ability of our method to recover when the instrument moves out-of-view of the camera. The trajectories of the tracked centre and the precision plots are shown in Figures 4.13. The same effect occurs in Dataset III where the 3D only tracker loses tracking after occlusion and does not recover. We show some tracking examples in Figure 4.14. After the instrument re-enter the frame view, the model never recover for pure 3D tracking (Figure 4.14 (c) and (d)). With the assistance of our 2D tracker, our 3D tracker can track the sequence well.

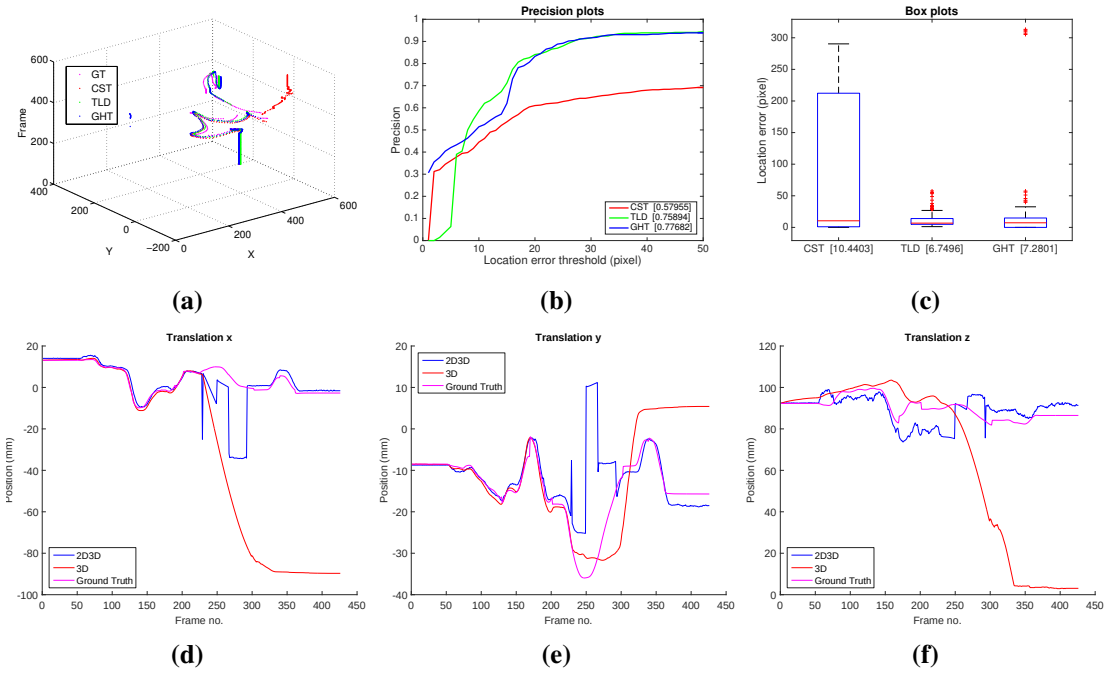


Figure 4.13: Performance comparison for Dataset III, which contains out-of-view occlusions between frames 325-350.

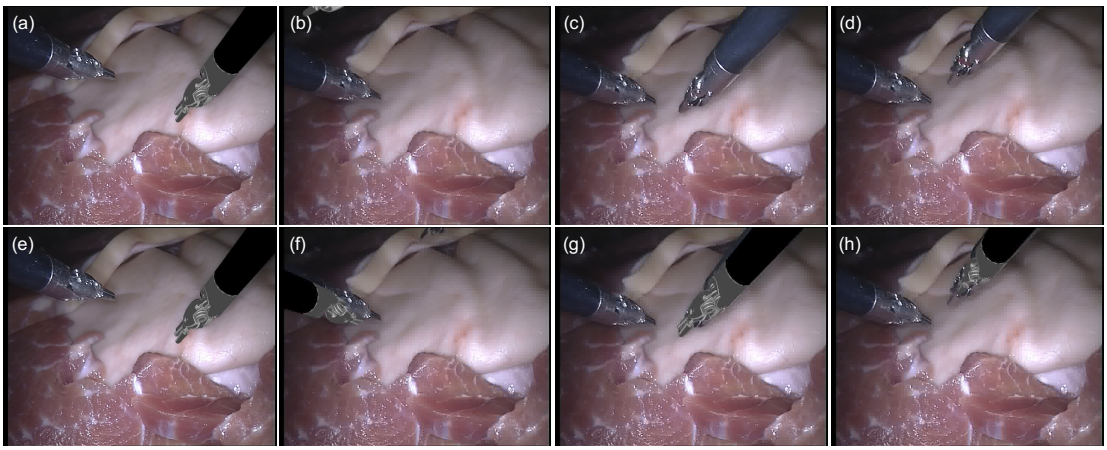


Figure 4.14: Frame examples of performance comparison between 2D3D tracking and pure 3D tracking under out-of-view. (a-d) Pure 3D tracker result; (e-h) 2D3D tracker result.

Long Term Tracking We construct an extended sequence (Dataset IV) of over 4000 frames to demonstrate the capability of our method to track the pose of the instrument in 3D without failing from drift. We display the results in Figure 4.15 where (a) shows the trajectories of tracked centre, (b) shows a precision plot for three 2D methods over the whole sequence, (c) shows a precision plot for three 2D methods over frames where all methods report a positive detection, (d) shows a box plot for the three 2D methods, (e-g) show the 3D trajectory of the proposed 2D3D tracker compared with using the 3D tracker directly. The figures show that our method is capable of reliable long term tracking although it does exhibit interesting failure cases. On Dataset IV our method fails to discriminate between out-of-view occlusions and out-of-plane based appearance changes which results in non-detected output when the tracked

points are rotated out of the field of view or such that the appearance of the patch changes beyond recognition. To display quantitative precision plot results in cases where each 2D detection method has some false non-detections, we display 2 types of plot, one where we display the results from the whole sequence where we set an infinite distance for missed detections (Figure 4.15b) and one where we only consider frames where all of the 2D tracking methods report a detection (Figure 4.15c) .

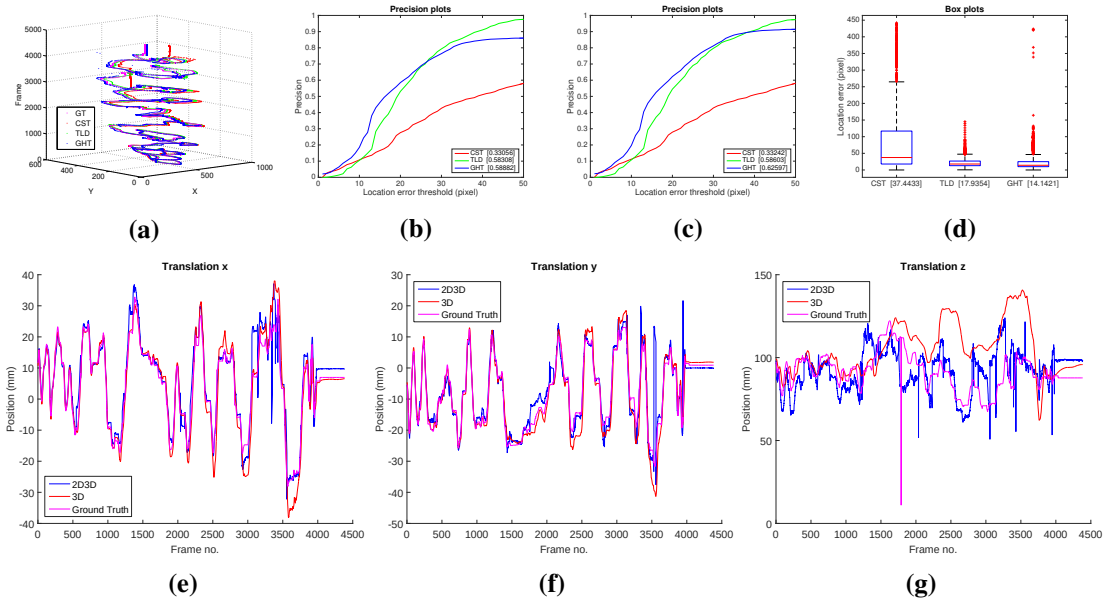


Figure 4.15: Performance comparison for the extended tracking sequence, Dataset IV.

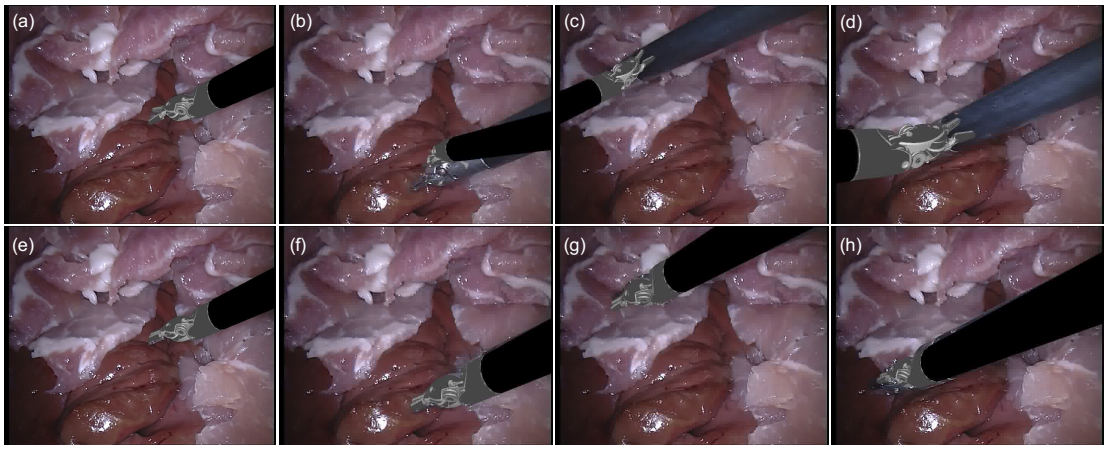


Figure 4.16: Frame examples of performance comparison between 2D3D tracking and pure 3D tracking for long term. (a-d) Pure 3D tracker result; (e-h) 2D3D tracker result.

The results of the above sequences show that the CST tracker lacks the ability to recover from occlusions or out-of-view situations compared with the TLD tracker and our GHT tracker. Our GHT tracker has the highest AUC score among the three trackers, which means our method can handle various occlusion or out-of-view challenges. In Table 4.2, compared to pure 3D tracker, our tracker has lower tracking error for all the sequences.

	Dataset I	Dataset II	Dataset III	Dataset IV
2D3D	3.70 ± 2.28	16.23 ± 11.83	8.29 ± 11.29	11.54 ± 7.94
3D Only	4.76 ± 3.28	38.47 ± 32.11	51.37 ± 52.10	16.79 ± 14.88

Table 4.2: Numerical results for the 3D tracking for each of the *ex vivo* sequences. Each value shows the mean error (mm) of the translation error for our 2D3D method and for the 3D only tracking.

4.9.2 *In Vivo* Experiments

We additionally qualitatively validate our method using robotic video data [134]. Example images showing our method performing detection on these images are shown in Figure 4.17. This *in vivo* sequence shows that our method is capable of tracking through complex surgical images even when the instrument undergoes articulation, which our method does not explicitly model.

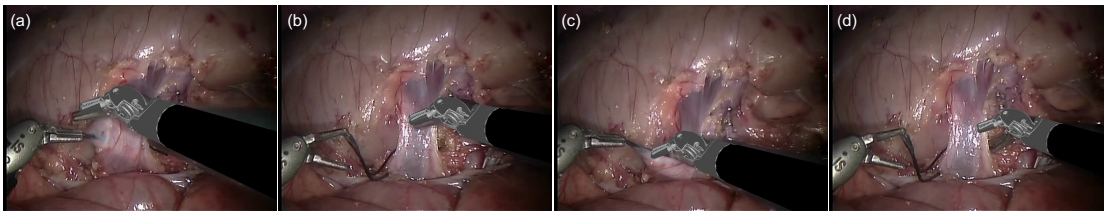


Figure 4.17: Frames showing an instrument tracked through an *in vivo* sequence. (a-c) demonstrate good accuracy whereas in (d) a failure mode for our algorithm is exhibited where poor classification on the instrument body causes the 3D tracked to fail to converge correctly.

4.10 Discussion

In this chapter, we presented a new method which combines the strengths of a novel 2D tracker with a pre-existing 3D tracking method [137] allowing us to robustly track surgical instruments through sequences that contain occlusions and challenging motion which cause the 3D tracker to fail. The 2D tracker is based on GHT and is updated with a global histogram probabilistic segmentation model. We quantitatively validate our method using *ex vivo* data collected from a dVRK controller and forward kinematics and additionally provide convincing qualitative validation on *in vivo* robot-assisted prostatectomy sequences. Our validation shows that our method provides state-of-the-art 2D tracking performance and significantly improves tracking accuracy in 3D. In the *ex vivo* sequences we restrict the motion of rigid 3D tracking as the method we use [137] does not model articulations of the instrument tip. Our extensive *ex vivo* validation demonstrates that our method is not only capable of tracking instruments over extended sequences but that it can also recover from tracking failures and occlusions, a feature that has not been demonstrated in any prior 3D tracking work in a MIS context. In terms of our 2D tracker, one of the limitations is that it is designed to handle only in-plane rotation and treat out-of-plane rotation as non-detection (e.g. in Dataset IV experiment in Section 4.9.1). This is partly because the objective of the 2D tracker in our application is to achieve re-detection and provide relatively accurate in-plane rotation estimation compared with its original pose in the first frame. Our model of the instrument model is based on the 2D spatial distribution of the keypoints with regards to the reference centre. Whenever there is out-of-plane rotation, the

distribution of the keypoints has changed in the image plane, therefore, the in-plane rotation θ is no longer valid. Future improvements could be use feedback from the 3D tracker to project into the image plane and update the 2D object model. Furthermore, we could focus around removing the requirement on a manual initialization. This can potentially be achieved with an enforced fixed position of the instrument while the 3D pose estimator converges to a correct solution.

Feature-based trackers have the advantage of detecting the target in a global search. But their tracking performance highly relies on the quality of feature detection and matching, which means they will have difficulty with targets which lack discriminative features. Even with a voting scheme to deal with the outliers, without enough correctly matched features, the algorithm cannot infer the object position. In the next chapter, we will tackle the problem using online learning techniques. Instead of treating the tracking problem as model fitting, it focuses on distinguishing between the target and the surrounding background by online classifiers. Recently, tracking-by-detection methods are widely used and dominating in object tracking, unlike generative methods, they utilize information not only from the target, but also from the background, and then find a decision boundary to separate the target from the surrounding.

Chapter 5

Online Tracking-by-Detection of Surgical Instruments

5.1 Introduction

The key components of a successful tracking algorithm combine a model to represent the object and how to update this object model over time as well as how to estimate the similarity between models. Recently, inspired by the success of object detection algorithms, the tracking-by-detection methods have been taking inspiration from advances in machine learning, such as structured output SVM [197, 5], boosting [77, 79], Gaussian process regression [198] and deep learning [101]. A typical tracking-by-detection algorithm treats the tracking as a classification task, it usually builds a classifier in the first frame to distinguish the tracked object from the background and updates this model over time with collected positive observations as well as with negative information. Under challenging conditions, objects usually undergo different transformations, such as deformation, scale change, occlusion or all at the same time, all those challenging factors make the accurate scale estimation quite difficult. Even with accurate scale prediction, it is inevitable that falsely labelled samples will appear and degrade the model because background information within the positive samples is falsely included which confuses the classifier and ultimately leads to drift or failure. Hare et al. [5] introduced Structured Output Tracking with Kernels (Struck) approach, which adopts a structured output SVM and circumvents the traditional collection of positive and negative samples by integrating the labelling procedure within the learning process. In recent benchmark [42] Struck has shown excellent tracking performance compared to prior work.

For object representation, recently patch-wise object descriptors have been exploited to represent the object appearance [2, 199, 200]. The bounding box is divided into cells or patches and low-level features are used to construct the features of these patches, which represent local structural information. A major challenge for tracking-by-detection methods is that the bounding box usually not only includes the object but also some background information. The background changes differently to the moving object and causes inaccurate information transfer through the model update. To address this problem, different methods have been proposed to decrease the effects of the background information such as assigning different weights based on the pixel spatial location or appearance similarity [51, 201, 202]. Kim et. al [2] exploited this concept by incorporating Random Walk with Restart simulations to assign weights to patches.

The simulations exploit the similarity between neighbouring patches and their relevance or self-similarity to the object appearance. Stationary distributions can be obtained to represent the probabilities that each patch belongs to either the foreground or the background. The patch weights are designed according to the likelihoods so that foreground patches would have relatively larger weights. Inspired by Kim et.al's work [2], we introduce a different weighting method to patches by incorporating a colour-based segmentation model. Previous papers have integrated a segmentation step into tracking [203, 196], but these methods are sensitive to segmentation results since they directly track the segmented object patches free from the constraints of bounding box. By applying a segmentation step to patch weights instead we manage to enhance performance and avoid this sensitivity.

5.2 Probabilistic Segmentation Model for Patch Weighting

We used the patch-wise descriptor [2, 199] to represent the appearance of the object. In frame t, the bounding box Ω is evenly decomposed into n_φ non-overlapping patches $\{\varphi_i\}_{i=1:n_\varphi}$, then the descriptor $\mathbf{\Phi}_{\Omega,t}$ is constructed by concatenating the low-level feature vectors of all the patches in their spatial order. Since background information is potentially included in the bounding box, we would like to incorporate an global probabilistic segmentation model [196, 195] to assign weights to the patches based on their colour appearance.

$$\mathbf{\Phi}_{\Omega,t} = [w_{1,t}\phi_1^T, \ldots, w_{n_\varphi,t}\phi_{n_\varphi}^T]^T \tag{5.1}$$

where w_i is the weight of the feature vector ϕ_i of the i-th patch φ_i. The global segmentation model is based on colour histogram by using a recursive Bayesian formulation to discriminate foreground (object) and background.

Let $y_{1:t}$ be the colour observation of a pixel from frame 1 to t, the foreground probability of that pixel at frame t are based on the tracked results from previous frames

$$p(c_t = 1|y_{1:t}) = Z^{-1} \sum_{c_{t-1}} p(y_t|c_t = 1)p(c_t = 1|c_{t-1})p(c_{t-1}|y_{1:t-1}) \tag{5.2}$$

where c_t is the class of the pixel at frame t: 0 for background, and 1 for foreground, and Z is a normalization constant to keep the probabilities sum to 1. The transition probabilities for foreground and background $p(c_t|c_{t-1})$ where $c \in \{0, 1\}$ are empirical choices as in [196]. The distributions $P(y_t|c_t)$ are modelled with colour histograms. The foreground histogram $p(y_t|c_t = 1)$ and the background histogram $p(y_t|c_t = 0)$ are initialized from the pixels inside the bounding box and from those which are surrounding the bounding box (with some margin between) in the first frame, respectively. For the following frames, the colour histogram distributions are updated using the tracked result.

$$\begin{aligned} p(y_t|c_t = 1) =& \delta p(y_t|y_t \in \Omega_t)) \\ &+ (1 - \delta)p(y_{t-1}|c_{t-1} = 1) \end{aligned} \tag{5.3}$$

where $0 \leq \delta \leq 1$ is the model update factor. The linear update with fixed update factor δ decides the adaption speed of the target colour histogram distribution in the sense that the

contribution of a specific frame decreases exponentially the further it recedes into the past [204]. We assume that the colour model of the target does not change dramatically between frames, so we empirically set the value of the update factor δ as 0.1, which is robust enough for appearance adaption and sudden occlusion. Ω_t represents the tracked bounding box in frame t. Since the tracked bounding box may contain background pixels, instead of treating every pixel equal in [196], we weight the pixels based on the weight of the patch where it is located. Patches with higher weight are more likely to contain object pixels and vice versa. So the colour histogram update for colour observation y_t of current frame t is defined as

$$p(y_t|y_t \in \Omega_t) = \frac{\sum_{i=1}^{n_\varphi} w_{i,t-1} N_{y_t \in \varphi_{i,t}}}{\sum_{i=1}^{n_\varphi} w_{i,t-1} \sum_{x_t} N_{x_t \in \varphi_{i,t}}} \tag{5.4}$$

where $N_{y_t \in \varphi_{i,t}}$ represents the number of pixels with colour observation y_t in the i-th patch $\varphi_{i,t}$ in frame t, and x_t represents any colour observation in frame t, so the denominator means the weighted number of all the pixel colour observations in the bounding box Ω_t. The distributions are updated based on the patch weight, which could reinforce the colour distribution of the object model.

We initialize the weight $w_{i,1}$ for all the patches as 1 at the first frame, and then they are updated based on the segmentation model

$$w_{i,t} = \delta \bar{w}_{i,t} + (1 - \delta) w_{i,t-1} \tag{5.5}$$

$$\bar{w}_{i,t} = \frac{\varpi_{i,t}}{\max_{1 \le i \le n_\varphi} \varpi_{i,t}} \tag{5.6}$$

$$\varpi_{i,t} = \frac{\sum_{x_t} p(x_t|c_t = 1) N_{x_t \in \varphi_{i,t}}}{\sum_{x_t} N_{x_t \in \varphi_{i,t}}} \tag{5.7}$$

where $\varpi_{i,t}$ denotes the average foreground probability of all pixels in the patch $\varphi_{i,t}$ in the current frame t, it is normalized so the highest weight update $\bar{w}_{i,t}$ equals 1. The patch weight $w_{i,t}$ is then updated gradually over time. We omit the background probability $p(c_t = 0|y_{1:t})$ discussion here since it is similar to Equation 5.2. Notice that unlike the weighting strategy in [2, 199] by analyzing the similarities between neighbouring patches, our patch weighting method is simple and straightforward to implement, the weight update for each patch is independent from each other, and only relies on the colour histogram based segmentation model. We show examples of the patch weight evolvement in Figure 5.1. The patch weight thumbnails are displayed on the top corner of each frame, which represent the deformation of the object over time. Notice that the size of the thumbnail varies in the same sequence, because we adapt scale estimation in our tracking framework, which will be discussed in Section 5.3 and Section 5.4. For example, in "Skiing", the size of the object is small, which make it harder to be tracked, our framework can adapt the bounding box according to the object, also the patch weight indicates the objectness information in the bounding box, which suppress the background information efficiently. In "Tiger2", the object is occluded between the 250th to 254th frame, since we update the segmentation model based on the previous patch weight, and in turn the segmentation model facilitates updating the weight patches. This co-training strategy enhances the weight contrast between foreground and occluded patches. Also, even there are illumination changes (the 273th

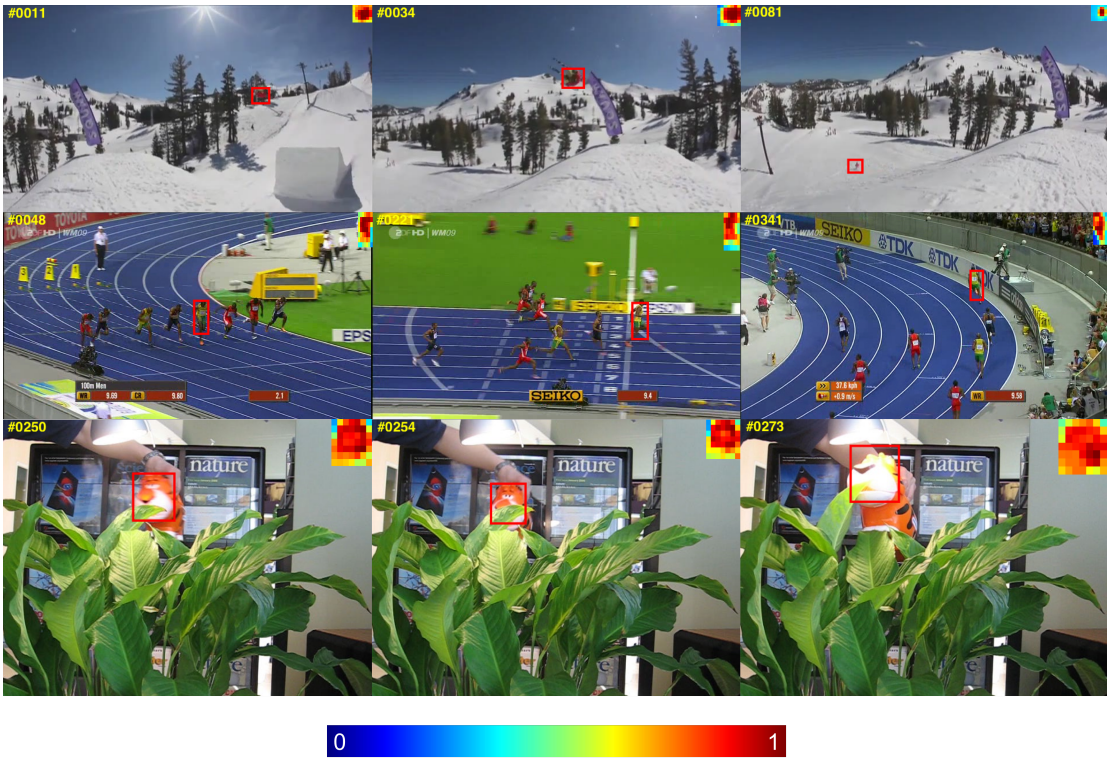


Figure 5.1: Example patch weights are shown for the highlighted bounding box displayed in the top corner of the image. The colour bar indicates the weight where 0 is considered more background and 1 is considered to support foreground.

frame), the patch weighting is robust to distinguish the object from the background. We evaluate the performance in section 5.5, which shows that our weighting strategy is adequate to represent the object appearance over time.

5.3 Scale Estimation

The tracked object often undergoes complicated transformations during tracking, for example, deformation, scale variations, occlusion et. al as shown in Figure 5.2. Fixed-scale bounding box estimation is ill-equipped to capture the accurate extents of the object, which would degrade the classifier performance by providing samples which are either partial cropped or include background information.

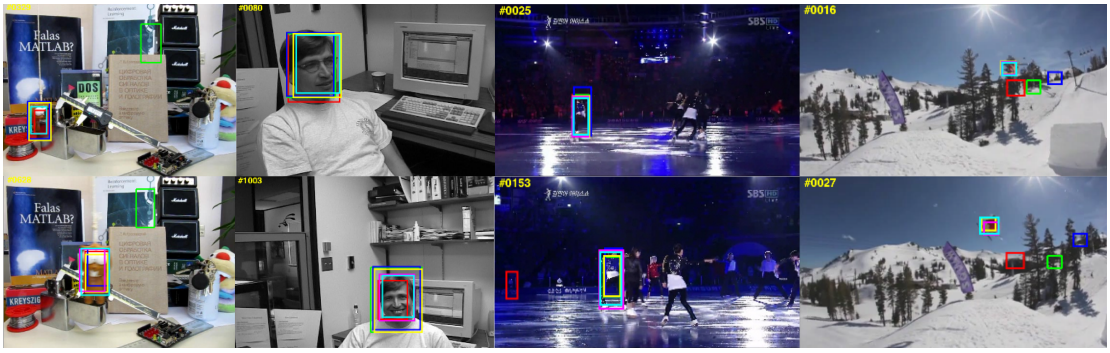


Figure 5.2: Examples of object undergo challenging transformations for tracking, inclusion of background information or partial object within the bounding box usually degrade the classifier.

In the original Struck [5] tracking framework, when locating the object in a new frame, all the bounding box candidates are collected within a searching window, and the bounding box with the maximum classification score is selected to update the object location. Rather than making a suboptimal decision by choosing from fixed-scale samples, we augment the training sample pool with multi-scale candidates. Obviously, the scales of the augmented samples are critical. We consider two complementary strategies that handle both incremental and abrupt scale variations.

Firstly, to deal with relatively small scale changes between frames, we build a scale set S_r

$$S_r = \{s|s = \lambda^m s_{t-1}\} \quad m \in [-\frac{n_r - 1}{2}, \ldots, \frac{n_r - 1}{2}] \tag{5.8}$$

where λ is a fixed value which is slightly larger than 1.0, n_r is the scale number in the scale set S_r. s_{t-1} is the scale of the object in frame $t-1$ compared with the initial bounding box in the first frame. Considering object scale usually does not vary too much between frames, scale set S_r includes scales which are close to the previous frame.

Secondly, when object undergoes abrupt scale changes between frames, scale set S_r is unable to keep pace with the speed of the scale variations. To address this problem, we build an additional scale set S_p by incorporating KLT tracker [205], which helps us estimate the scale change explicitly. We pick the top n_{pt} strongest corner points from each patch in the bounding box Ω_{t-1} of frame $t-1$ using Shi-Tomasi method [206], and tracked all these points in the next frame t. With sufficient well-tracked points, we can estimate the scale variation between frames by comparing the distance changes of the tracked point pairs. We illustrated in Figure 5.3. Let

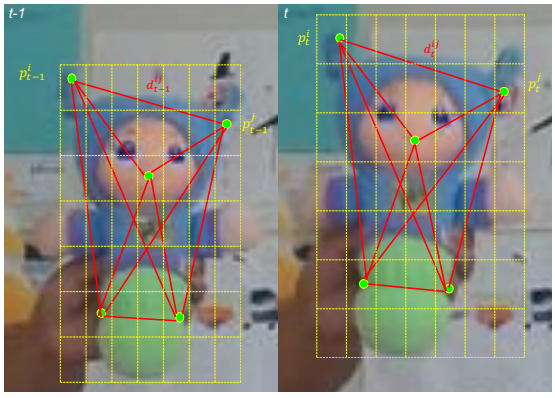


Figure 5.3: Illustration of the scale estimation by using the KLT tracker. Corner Points on the patches are picked in frame $t-1$, and are tracked in the next frame t by the KLT tracker, the distance ratio of point pairs (p^i, p^j) between two frames are used for scale estimation.

p^i_{t-1} denotes one picked point in the previous frame $t-1$ and its matched point p^i_t in the current frame t. We compute the distance d^{ij}_{t-1} between point-pair (p^i_{t-1}, p^j_{t-1}), and the distance d^{ij}_t between the matched point-pair (p^i_t, p^j_t).

For all the matched point pairs, we compute the distance ratio between the two frames

$$V = \{s|s = d^{ij}_t / d^{ij}_{t-1}\} \quad i \neq j \tag{5.9}$$

where V is the set with all the distance ratios. We sort V by value and pick the median element $s_p = V_{sorted}(\frac{n}{2})$ as the potential scale change of the object. To make the scale estimation more robust, we uniformly sample the scales ranging between $[1, s_p]$ or $[s_p, 1]$ to construct the scale set S_p.

$$S_p = \{s | s = 1 + i\frac{s_p - 1}{n_p - 1}\} \quad 0 \leq i < n_p \tag{5.10}$$

where n_p is the scale number in the scale set S_p. When the object is out-of-view, occluded or abruptly deforms, the ratio of well-tracked points will be low. In that case, the estimation from the KLT tracker will be unreliable. In our implementation, when the ratio is lower than 0.5, we then set $s_p = 1$, therefore the scale set S_p will only add samples with the previous scale into the candidate pool. Only when there are enough points well tracked, the estimation from the KLT tracker will be trusted. We fuse these two complementary scale sets S_r and S_p into $S_f = S_r \cup S_p$ to enrich our sample candidate pool. To show the effectiveness, we evaluate our proposed tracker in section 5.5 with or without scale set S_p estimated by the KLT tracker.

5.4 Tracking Framework

We incorporate PAWSS into the Struck [5]. The algorithm relies on an online structured output SVM learning framework which integrates the learning and tracking. It directly predicts the location displacement between frame, avoiding the heuristic intermediate step for assigning binary labels to training samples, which achieves top performance in the OTB dataset [42].

Given the bounding box Ω_{t-1} in the previous frame $t-1$, sample candidates are extracted in a searching window r_w, which centres at the Ω_{t-1} in the current frame t, unlike other tracking-by-detection approaches, we adapt a two-level sampling strategy. On the first level, all the bounding box samples are extracted with fixed-scale s_{t-1}, on the second level, multi-scale samples are extracted to enrich the sample pool.

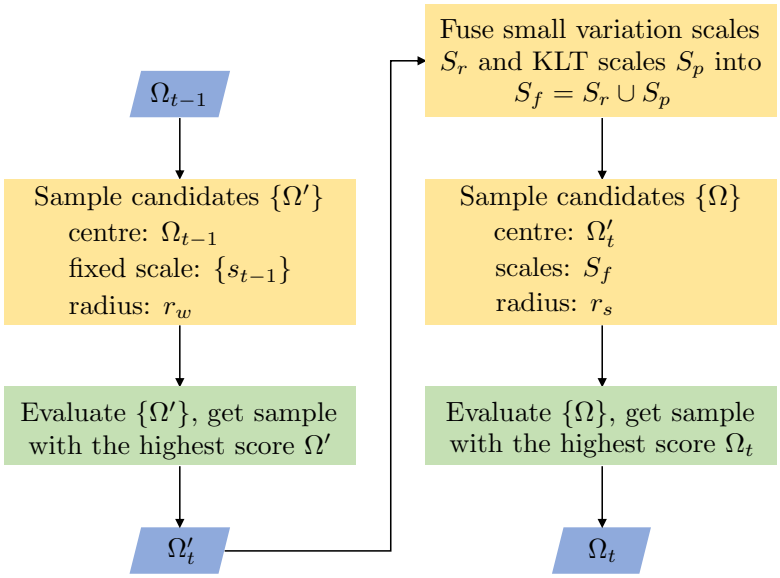


Figure 5.4: Two-level sampling strategy workflow

The detailed two-level sampling strategy workflow is shown in Figure 5.4. Assume that we have the tracking bounding box Ω_{t-1} at time $t-1$. First, the searching window is chosen at the

same as above centred at the Ω_{t-1} with a radius of r_w, since we have the second level to make the final decision, rather than extracting sample per pixel, we extract samples at a down-sample factor of 2, which could decrease the candidate number by 4, then the weighted patch-wise descriptor of each candidate is constructed, and we select the bounding box with the maximum classification score Ω' not as the final decision, but as the search centre for our second level. After this step, the rough location of the object is narrowed into a smaller area. Like discussed in Section 5.3, given the scale s_{t-1} in the previous frame $t-1$, to handle small scale variation between frames, we construct the scale set S_r, which includes scales which are close to s_{t-1}. Additionally, to deal with potential abrupt scale changes, we random pick n_{pt} points from each patch of the bounding box Ω_{t-1}, and pass all these points to the KLT tracker to generate the scale set S_p. This scale set is estimated explicitly by the KLT tracker and facilitates to augment the scale estimation. Then we fuse the scale two complementary scale sets S_r and S_p into S_f to extract bounding box candidates. We set a smaller search window with search radius of r_s, centreing at the bounding box Ω' selected in the first level, and we construct multiple candidates for each pixel within the search window. The scales of candidates at one pixel are set as scales in the fused scale set S_f. We then evaluate all the multi-scale samples and select the bounding box sample with the maxiumn score Ω_t as the final location of the object. For multiple bounding box samples with the same scores, the sample whose scale is closer to 1.0 is selected to prevent potential gradual shrinking or enlargement of the bounding box.

Then, the classifier, the colour-based segmentation model and the weights of all patches are updated as discussed in Section 5.2. Finally, the whole process starts at the next frame. Additionally, to prevent introducing potential corrupt samples to the classifier, we use the same updating strategy introduced in [2], the classifier only updates when the similarity between the tracked object and the positive support vectors are above certain threshold η.

5.5 Experiments and Results

5.5.1 Implementation Details

Our proposed algorithm is implemented in C++ and performs at about 7 frames per second with an i7-2.5GHz CPU without any optimisation. For structured output SVM, different kernels can be implemented and even combined for evaluating patch similarity, such as linear, Gaussian or intersection kernel. From Struck [5], Naïve kernel combinations do not give a significant performance gain, so in our evaluation we use the simplest kernel function, the linear kernel $k(\mathbf{\Phi}_{\Omega_1}, \mathbf{\Phi}_{\Omega_2}) = \mathbf{\Phi}_{\Omega_1}^T \mathbf{\Phi}_{\Omega_2}$, in which $\mathbf{\Phi}_{\Omega_1}$ and $\mathbf{\Phi}_{\Omega_2}$ are feature vectors for bounding box Ω_1 and Ω_2, respectively. The parameters are empirically set as $\delta = 0.1$ in Equation 5.3 and Equation 5.5, $\lambda = 1.003$ in Equation 5.8, the scale numbers of the scale set are $n_r = n_p = 11$. The number of extracted points from each patch $n_{pt} = 5$. The updating threshold for classifier is set as $\eta = 0.3$. For each sequence, we scale the frame to make sure the minimum side length of the bounding box is larger than 32 pixels, and the search window radius r_w is fixed to $(W + H)/2$, where W and H represents the width and height of the scaled bounding box, respectively, the search window radius r_s is fixed to 5 pixels. The patch number affects the tracking performance, too many patches increase the computation and too less patches do not robustly reflect the local appearance of the object. We tested different patch numbers and selected $n_\varphi = 49$ in

our implementation to strike a performance balance.

5.5.2 OTB Dataset

The OTB dataset [42] includes 50 commonly used sequences with fully annotations. The sequences are also tagged with 11 attributes, which represent the challenging aspects for tracking such as illumination variation, occlusion, deformation et al. The tracking performance is quantitatively evaluated using both PR and SR, as defined in [42]. PR / SR scores are depicted using precision plot and success plot, respectively. The precision plot shows the percentage of frames whose tracked centre is within certain Euclidean distance (20 pixels) from the centre of the GT. Success plot computes the percentage of frames whose intersection over union overlap with the GT annotation is within a threshold varying between 0 and 1, and the AUC is used for SR score. In [42], the authors mentioned that the rankings of certain trackers in the success plots are different from the rankings in the precision plot. SR is more accurate than PR since the AUC score represents the overall performance rather than the score at one threshold. So the rankings are based on success plots, and the precision plots are used as auxiliary. In the following evaluation, to evaluate the effectiveness of incorporating the scale set proposed by the KLT tracker, we provide two versions of our tracker as PAWSSa and PAWSSb: PAWSSa only includes scale set S_r, while PAWSSb includes both S_r and S_p for scale estimation.

Comparison Using Different Features Selecting the right features to describe the object appearance plays a critical role in tracking. The most desirable feature property is its uniqueness so that the object can be distinguished from the background. Raw intensities or colour features are usually used for histogram-based appearance representations, while edge or gradient information are less sensitive to illumination changes. Generally, many tracking approaches use a combination of these diverse features to represent the object [5, 79, 207]. To evaluate the performance of our proposed approach, we tested different low-level features such as HSV colour, RGB colour, the combination of colour and gradient features (HSV+G, RGB+G) for constructing the descriptor in Table 5.1. The RGB histogram is 24-dimensional with 8 bins for each channel, and the HSV colour histogram is 20-dimensional including 8 bins for H and S channels respectively and 4 separate bins for V channel. The gradient histogram is 16-dimensional signed gradients ranging from 0 to 360 degrees. We also compared our tracker PAWSSa and PAWSSb with Struck [5] and SOWP [2] in Table 5.2.

From Table 5.1 and Table 5.2, we observe: First, augmenting colour with gradient histogram improves the tracking performance by providing diverse structural information of the object. In our experiments, the descriptor comprising the combination of HSV colour and gradient features achieves the best results, we would use this setting in the following evaluation; Second, by using a simple patch weighting strategy and training with adaptive scale samples, our tracker achieves 36.7% gain in PR and 36.9% gain in SR over Struck. Compared with SOWP [2], the performance shows that our tracker provides comparable PR scores, and higher SR score. PAWSSa tracker improves the SR score by 2.6% considering gradually small changes between frames, PAWSSb improves the SR score by 4.8% by incorporating scales estimated by the external KLT tracker.

Comparison with State-of-the-art Trackers We use the evaluation toolkit provided by Wu

	PAWSSa	PAWSSb
HSV	0.731 / 0.528	0.742 / 0.545
RGB	0.764 / 0.552	0.749 / 0.544
RGB+G	0.838 / 0.605	0.840 / 0.607
HSV+G	**0.889 / 0.635**	**0.897 / 0.649**

Table 5.1: The performance of the proposed algorithm compared with different low-level features. PAWSSa and PAWSSb tracker represents our tracker without and with the KLT tracker, respectively.

	Struck	SOWP	PAWSSa	PAWSSb
PR	0.656	0.894 (36.3%)	0.889 (35.5%)	**0.897 (36.7%)**
SR	0.474	0.619 (30.6%)	0.635 (34.0%)	**0.649 (36.9%)**

Table 5.2: The performance of the proposed algorithm and the SOWP tracker [2] compared with the Struck tracker [5]

et al. [42] to generate the plots for the one pass evaluation (OPE) of the top 10 algorithms. The toolkit includes 29 benchmark trackers, besides that we also include SOWP tracker. The precision plot and success plot are demonstrated in Figure 5.5. It is shown that our proposed tracker PAWSSb achieves the best PR/SR scores among all trackers, with a 36.7% gain in PR over Struck and a 30.1% gain in SR over SCM.

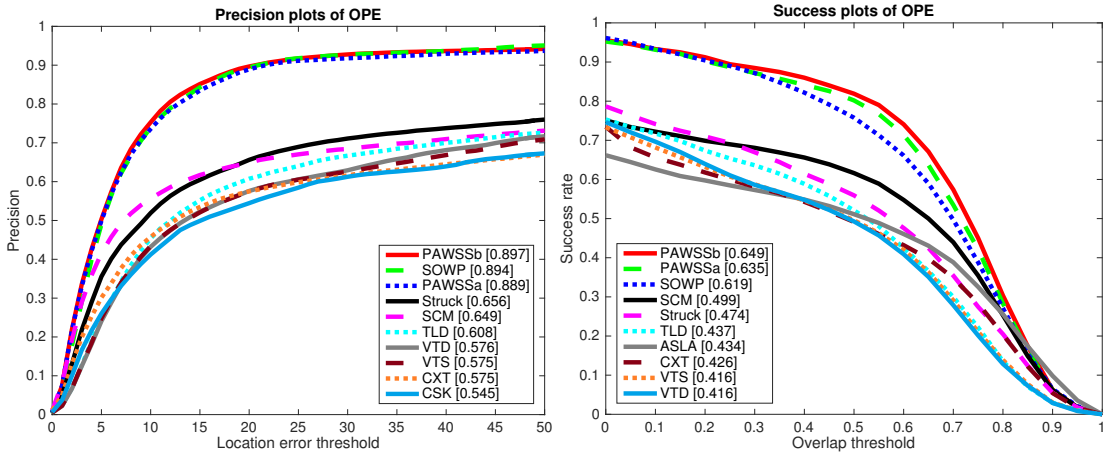


Figure 5.5: Comparison of the precision and success plots on the OTB with the top 10 trackers; the PR scores are illustrated with the threshold at 20 pixels and the SR scores with the AUC in the legend.

In Table 5.3, the PR/SR scores of PAWSS compared with the benchmark trackers according to different challenging attributes are shown in details. Our proposed trackers provide the best PR/SR score with in every attribute. We list the performance gain of each attribute compared with the second-best tracker (excluding our trackers) in the table. As is shown, PAWSS especially excels when the object undergoes deformation or background cluster. For deformation, it achieves 59.4% gain over PR and 53.6% gain over SR. When the background near the target has similar colour or texture as the target, PAWSS can still distinguish the object from the background, which achieves 46.8% in PR and 41.3% in SR.

In Table 5.4, PAWSS trackers are also compared with the state-of-the-art trackers: MEEM [210], FCNT [101] and SOWP [2] et. al. In the table, the best and the second-best

	ASLA [103]	CXT [208]	CSK [6]	VTD [76]	VTS [209]	TLD [3]	SCM [4]	Struck [5]	PAWSSa	PAWSSb
IV (25)	0.517 / 0.429	0.501 / 0.368	0.481 / 0.369	0.557 / 0.420	0.573 / 0.429	0.537 / 0.399	0.594 / 0.473	0.558 / 0.428	0.860 / 0.616	**0.880(48.1%) / 0.648(37.0%)**
SV (28)	0.552 / 0.452	0.550 / 0.389	0.503 / 0.350	0.597 / 0.405	0.582 / 0.400	0.606 / 0.421	0.672 / 0.518	0.639 / 0.425	**0.849** / 0.564	**0.849(32.9%) / 0.577(11.4%)**
OCC (29)	0.460 / 0.376	0.491 / 0.372	0.500 / 0.365	0.545 / 0.403	0.534 / 0.398	0.563 / 0.402	0.640 / 0.487	0.564 / 0.413	0.859 / 0.618	**0.872(36.3%) / 0.634(30.2%)**
DEF (19)	0.445 / 0.372	0.422 / 0.324	0.476 / 0.343	0.501 / 0.377	0.487 / 0.368	0.512 / 0.378	0.586 / 0.448	0.521 / 0.393	0.908 / 0.656	**0.934(59.4%) / 0.688(53.6%)**
MB (12)	0.278 / 0.258	0.509 / 0.369	0.342 / 0.305	0.375 / 0.309	0.375 / 0.304	0.518 / 0.404	0.339 / 0.298	0.551 / 0.433	**0.786** / 0.593	0.783 (42.1%) / **0.603(39.3%)**
FM (17)	0.253 / 0.247	0.515 / 0.388	0.381 / 0.316	0.352 / 0.302	0.353 / 0.300	0.551 / 0.417	0.333 / 0.296	0.604 / 0.462	0.784 / 0.572	**0.792(31.1%) / 0.587(27.1%)**
IPR (31)	0.511 / 0.425	0.610 / 0.452	0.547 / 0.399	0.599 / 0.430	0.579 / 0.416	0.584 / 0.416	0.597 / 0.458	0.617 / 0.444	**0.860** / 0.594	0.852(38.1%) / **0.600(31.0%)**
OPR (39)	0.518 / 0.422	0.574 / 0.418	0.540 / 0.386	0.620 / 0.434	0.604 / 0.425	0.596 / 0.420	0.618 / 0.470	0.597 / 0.432	0.898 / 0.623	**0.901(45.3%) / 0.635(35.1%)**
OV (6)	0.333 / 0.312	0.510 / 0.427	0.379 / 0.349	0.462 / 0.446	0.455 / 0.443	0.576 / 0.457	0.429 / 0.361	0.539 / 0.459	0.771 / 0.611	**0.828(43.8%) / 0.645(40.5%)**
BC (21)	0.496 / 0.408	0.443 / 0.345	0.585 / 0.421	0.571 / 0.425	0.578 / 0.338	0.428 / 0.428	0.578 / 0.450	0.585 / 0.458	0.847 / 0.632	**0.859(46.8%) / 0.647(41.3%)**
LR (4)	0.156 / 0.157	0.371 / 0.312	0.411 / 0.350	0.168 / 0.177	0.187 / 0.168	0.349 / 0.309	0.305 / 0.279	0.545 / 0.372	**0.679 / 0.504**	0.669 (22.8%) / 0.500(34.4%)
Average (50)	0.532 / 0.434	0.575 / 0.426	0.545 / 0.398	0.576 / 0.416	0.575 / 0.416	0.608 / 0.437	0.649 / 0.499	0.656 / 0.474	0.889 / 0.635	**0.897(36.7%) / 0.649(30.1%)**

Table 5.3: Comparison of the PR/SR score in the OPE based on the 11 sequence attributes: illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background cluttered (BC) and low resolution (LR). The best results are shown in **bold**.

results are shown in red and blue colours respectively. Notice that in all the attribute field, our tracker achieves either the best or the second-best PR/SR scores among all the trackers. When the object undergoes scare variation, compared with SOWP, PAWSS achieves a performance gain of 10.3% in SR.

	DSST [211]	SAMF [212]	FCNT [101]	MTA [213]	TGPR [198]	DDCT [214]	MEEM [210]	SOWP [2]	PAWSSa	PAWSSb
IV (25)	0.727 / 0.534	0.735 / 0.563	0.830 / 0.598	0.738 / 0.547	0.687 / 0.486	0.665 / 0.499	0.778 / 0.548	0.842 / 0.596	**0.860 / 0.616**	**0.880 / 0.648**
SV (28)	0.723 / 0.516	0.730 / 0.541	**0.830** / 0.558	0.721 / 0.478	0.703 / 0.443	0.687 / 0.484	0.809 / 0.506	**0.849** / 0.523	**0.849 / 0.564**	**0.849 / 0.577**
OCC (29)	0.845 / **0.619**	0.716 / 0.534	0.797 / 0.571	0.772 / 0.563	0.708 / 0.494	0.723 / 0.534	0.815 / 0.560	**0.867** / 0.603	0.859 / 0.618	**0.872 / 0.634**
DEF (19)	0.813 / 0.622	0.660 / 0.510	0.917 / 0.644	0.851 / 0.622	0.768 / 0.556	0.804 / 0.602	0.859 / 0.582	**0.918 / 0.666**	0.908 / 0.656	**0.934 / 0.688**
MB (12)	0.651 / 0.519	0.547 / 0.464	**0.789** / 0.580	0.695 / 0.540	0.578 / 0.440	0.691 / 0.553	0.740 / 0.565	0.716 / 0.567	**0.786 / 0.593**	0.783 / **0.603**
FM (17)	0.663 / 0.515	0.517 / 0.435	0.767 / 0.565	0.677 / 0.524	0.575 / 0.441	0.685 / 0.534	0.757 / 0.568	0.744 / **0.575**	**0.784** / 0.572	**0.792 / 0.587**
IPR (31)	0.691 / 0.507	0.765 / 0.560	0.811 / 0.555	0.773 / 0.547	0.706 / 0.487	0.720 / 0.524	0.810 / 0.531	0.847 / 0.584	**0.860 / 0.594**	**0.852** / **0.600**
OPR (39)	0.763 / 0.554	0.733 / 0.535	0.831 / 0.581	0.777 / 0.557	0.741 / 0.507	0.726 / 0.518	0.854 / 0.566	0.896 / 0.615	**0.898 / 0.623**	**0.901 / 0.635**
OV (6)	0.708 / 0.609	0.515 / 0.459	0.741 / 0.592	0.612 / 0.534	0.495 / 0.431	0.622 / 0.524	0.730 / 0.597	**0.802 / 0.635**	0.771 / 0.611	**0.828 / 0.645**
BC (21)	0.708 / 0.524	0.694 / 0.517	0.799 / 0.564	0.795 / 0.592	0.761 / 0.543	0.660 / 0.502	0.808 / 0.578	0.839 / 0.618	**0.847 / 0.632**	**0.859 / 0.647**
LR (4)	0.459 / 0.361	0.497 / 0.409	**0.765 / 0.514**	0.579 / 0.397	0.539 / 0.351	0.526 / 0.411	0.494 / 0.367	0.606 / 0.410	**0.679 / 0.504**	0.669 / 0.500
Average (50)	0.777 / 0.570	0.737 / 0.554	0.856 / 0.599	0.812 / 0.583	0.759 / 0.539	0.762 / 0.557	0.840 / 0.570	**0.894** / 0.619	0.889 / **0.635**	**0.897 / 0.649**

Table 5.4: Comparison of the PR/SR score based on the 11 sequence attributes with state-of-the-art trackers in the OPE. For the descriptions of the challenging factors, refer to the caption of Table 5.3. The best and the second-best results are shown in **red** and **blue** colours respectively.

We show some tracking results in Figure 5.6 and Figure 5.7 with the top trackers including TLD [3], SCM [4], Struck [5], SOWP [2] and the proposed PAWSSa and PAWSSb. In Figure 5.6, five challenging sequences, "Matrix", "Ironman", "Skiing", "Skating1" and "Tiger2" are picked from the benchmark dataset, which include object scale variations, deformation, occlusion or background clusters. In "Matrix" and "Ironman", there are similar abstractions in the background and illumination changes dramatically, which makes the object extremely challenging to be tracked. PAWSS can not only track the object, but also estimates the scale accurately, as shown in the 76th frame and the 154th frame, respectively. In "Skiing", the object is particular small, PAWSS can adapt the bounding box scale when the object deforms. In "tiger2", the object is occluded during the sequence (for example in the 108th, 130th, 299th, 319th frame shown in the figure), with other trackers either losing tracking or slightly drifting away from the object, PAWSS tracks the object more reliably.

In Figure 5.7 we select five representative sequences with scale variations. In some of the sequences, the object gradually changes the scale between frames, for example in "Car4", "Walking" and "Dog1". SCM also estimates the scale of the object, but it fails to adapt the scale in the 1179th and the subsequent frames in "Dog1", while PAWSS adapts well. In "Lemming" and "Walking2" not only the scale changes, the object is also occluded, for example in the 323th frame in "Lemming", in the 375th frame when the object gets out of the obstacle,

Figure 5.6: Comparison of the tracking results of our proposed tracker PAWSS with SOWP [2] and three conventional trackers: TLD [3], SCM [4] and Struck [5] on some especially challenging sequences in the benchmark.

PAWSS can well track the object with scale variation, while the other trackers drift away. The result examples show that our weighting strategy and scale estimation can adapt to different scenarios, preventing the classifier from training with samples with partial object or with too much background information, which improves the tracking performance. The results show that our proposed tracking framework PAWSS can track the object robustly through sequence by using the weighting strategy to suppress the background information within the bounding box, and also by incorporating scale estimation allowing the classifier to train with adaptive scale samples.

Attribute-based Performance The plots for the performance of PAWSS and other trackers on the attribute subset sequences are shown from Figure 5.8 to Figure 5.11. (including ASLA [103], CPF [215], CSK [6], CXT [208], DFT [216], KMS [51], L1APG [217], LOT [218], LSK [219], MTT [220], OAB [79], SCM [4], SemiT [80], SOWP [2], Struck [5], TLD [3], template matching (TM-V) [221], VTD [76], VTS [222] and the proposed PAWSSa and PAWSSb.)

5.5.3 VOT Challenges Datasets

For completeness, we also validated our algorithm on VOT2014 (25 sequences) and VOT2015 (60 sequences) datasets. VOT datasets use ranking-based evaluation methodology: accuracy and robustness. Similar to SR rate for OTB dataset, the accuracy measures the overlap of the predicted result and the GT bounding box, while the robustness measures how many times the tracker fails during tracking. A failure is indicated whenever the tracker loses the target object which means the overlap becomes zero, and it will be re-initialized afterwards. All the trackers are evaluated, compared and ranked based on with respect to each measure separately in the

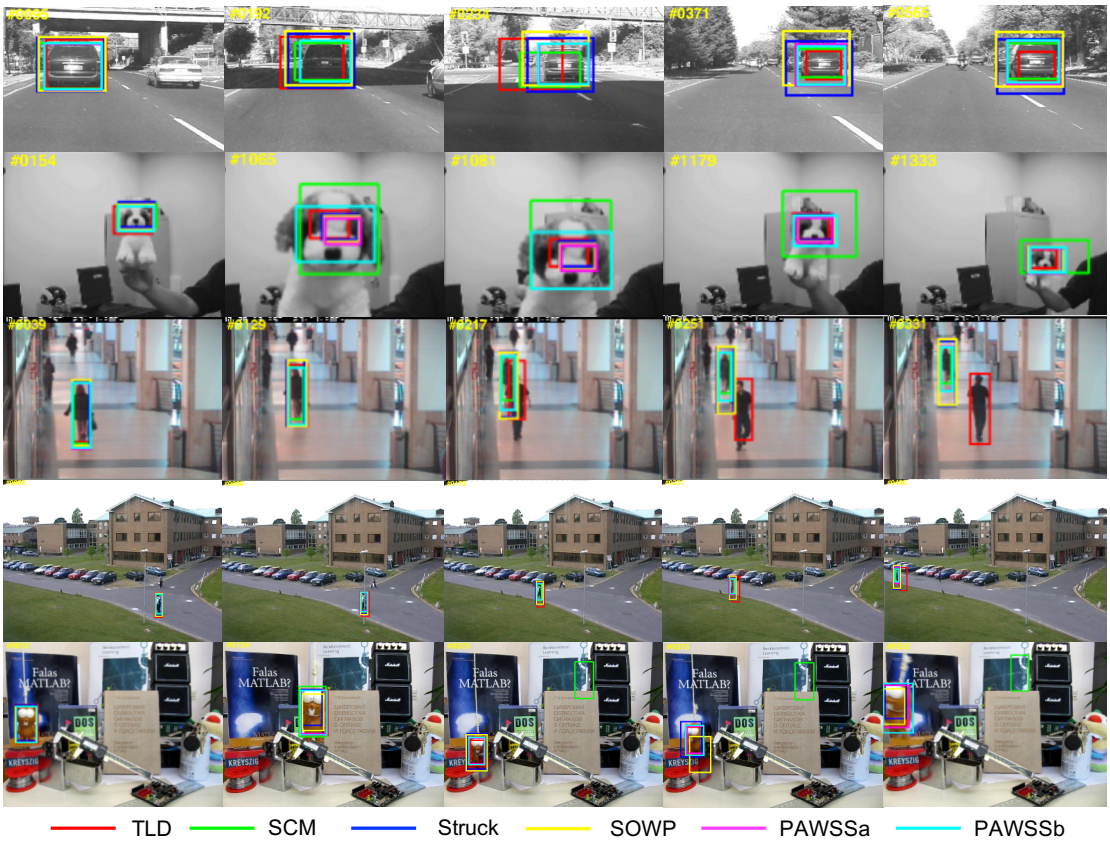


Figure 5.7: Comparison of the tracking results of our proposed tracker PAWSS with SOWP [2] and three conventional trackers: TLD [3], SCM [4] and Struck [5] on some sequences with scale variations in the benchmark.

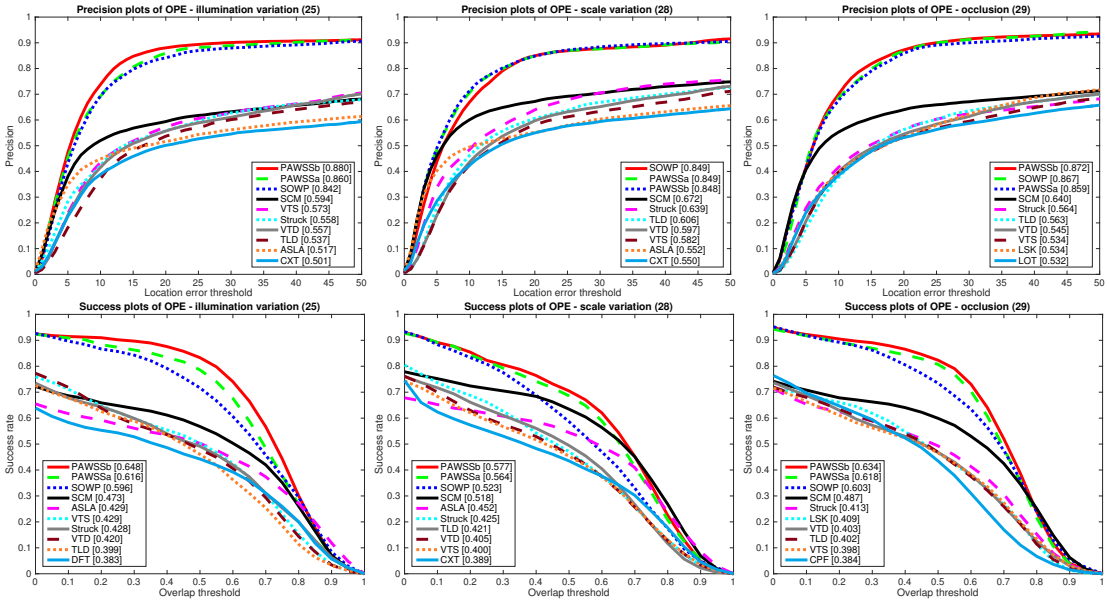


Figure 5.8: The plots for illumination variation, scale variation and occlusion sub-datasets. The number in the title is the number of sequences in that sub-dataset.

challenge.[1]

[1] http://www.votchallenge.net/

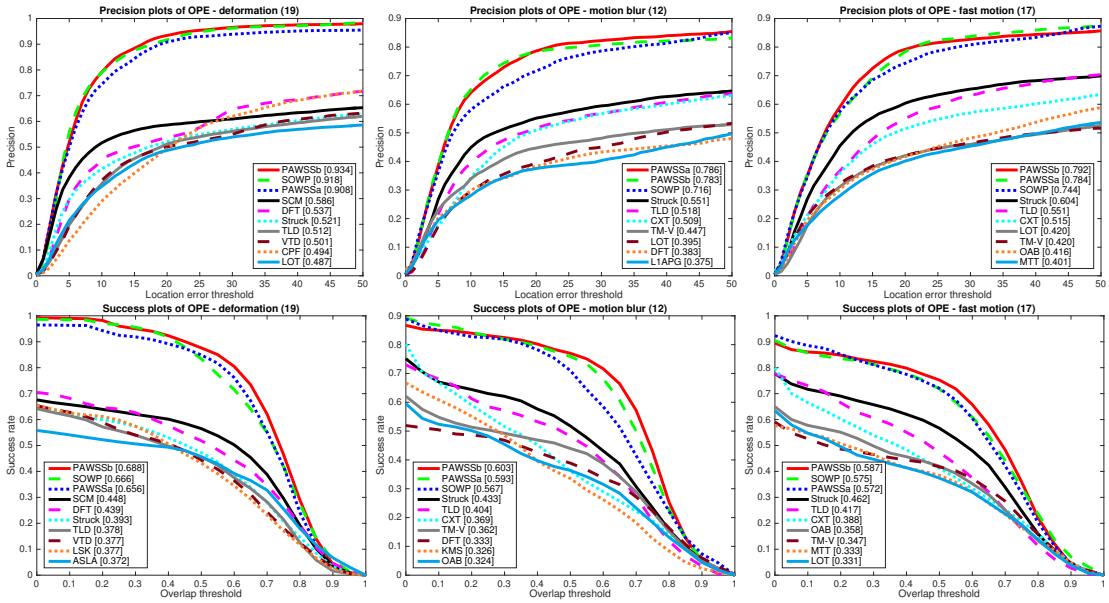


Figure 5.9: The plots for deformation, motion blur and fast motion sub-datasets.

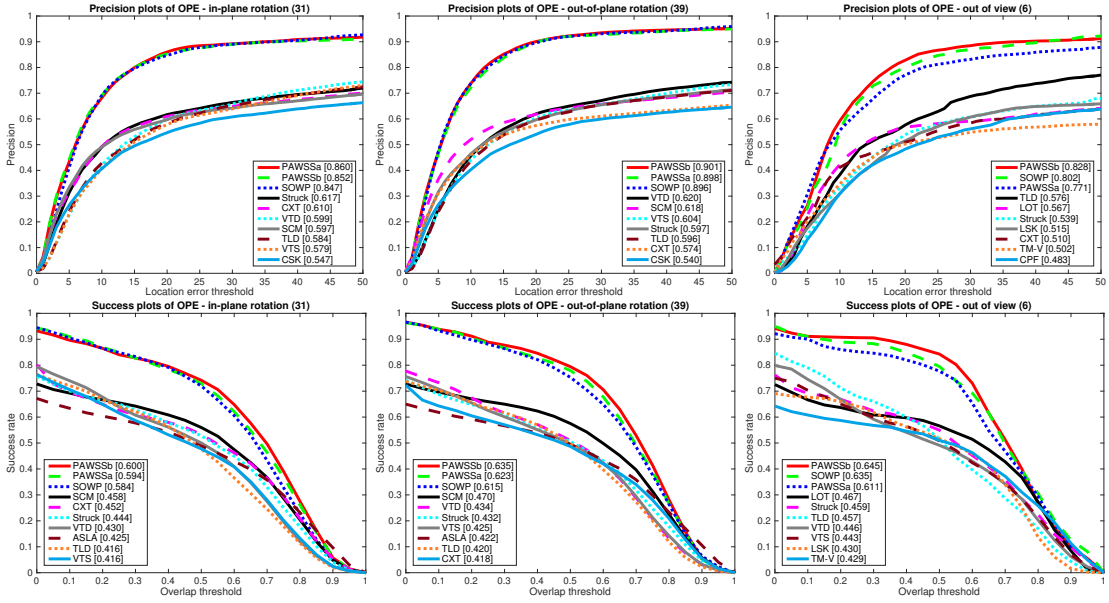


Figure 5.10: The plots for in-plane rotation, out-of-plane rotation and out-of-view sub-datasets.

VOT2014 The VOT2014 challenge includes two experiments: baseline experiment and region-noise experiment. In baseline experiment, a tracker runs on all the sequences by initializing with the GT bounding box on the first frame; while in the region-noise experiment, the tracker is initialized with a random noisy bounding box with the perturbation in the 10% of the GT bounding box size [223]. We compared our proposed method with the top three trackers among 38 trackers: DSST [211], SAMF [212], KCF [84]. From Table 5.5 and Figure 5.12, we can see that our tracker PAWSS has lower accuracy score but less failures. To eliminate the effect of achieving higher accuracy score by re-initialization step, we performed experiments without the re-initialization, shown in Table 5.6. The result show that PAWSS has the highest accuracy score without re-initialization, which means it is more robust than the other trackers.

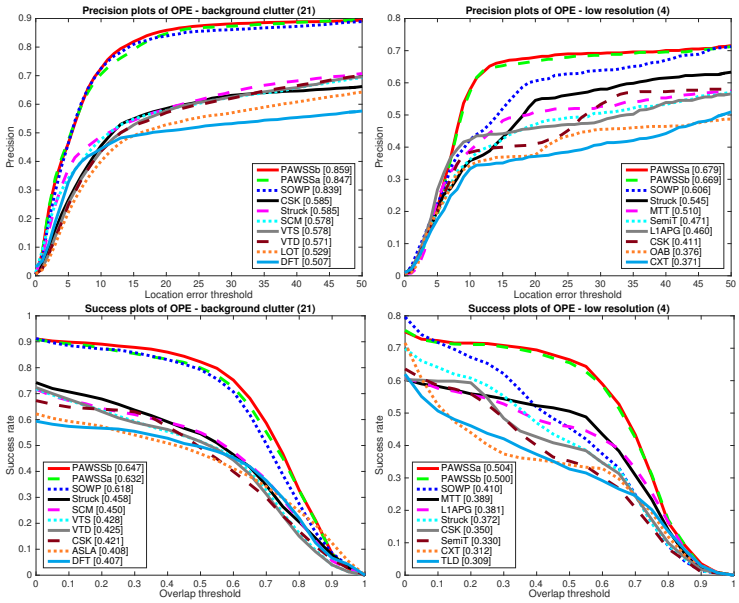


Figure 5.11: The plots for background clutter and low resolution sub-datasets.

	Baseline				**Region-noise**				Final Rank
	Accuracy		Robustness		Accuracy		Robustness		
	Score	Rank	Failure	Rank	Score	Rank	Failure	Rank	
DSST [211]	0.62	5.16	1.16	8.2	0.57	4.32	1.28	7.4	6.27
SAMF [212]	0.61	4.32	1.28	8.68	0.57	4.2	1.43	8.44	6.41
KCF [84]	0.62	3.68	1.32	8.68	0.57	4.84	1.51	9.00	6.92
PAWSSb	0.58	5.80	0.88	8.00	0.55	6.08	0.78	5.4	6.32

Table 5.5: VOT2014 results. The best and the second-best results are shown in **red** and **blue** colours respectively.

	Accuracy Score w/o	
	Baseline	Region-noise
DSST [211]	0.47	0.43
SAMF [212]	0.50	0.48
KCF [84]	0.40	0.36
PAWSSb	0.51	0.48

Table 5.6: VOT2014 without re-initialization results. The best and the second-best results are shown in **red** and **blue** colours respectively.

VOT2015 Finally, we evaluated and compared with 62 trackers on the VOT2015 dataset. The VOT2015 challenge only includes baseline experiment, and the ranking plots are shown in Figure 5.14.

In VOT2013 and VOT2014, average ranking measure is used to determine the performance of the trackers. Although average ranking has taken both accuracy and robustness measure into consideration, it is not theoretically representative as a concrete tracking performance. So expected average overlap measure which combines both per-frame accuracies and failures in a principled manner is introduced for VOT2015 [224]. Compared with the average ranking, expected overlap has a more clear practical interpretation.

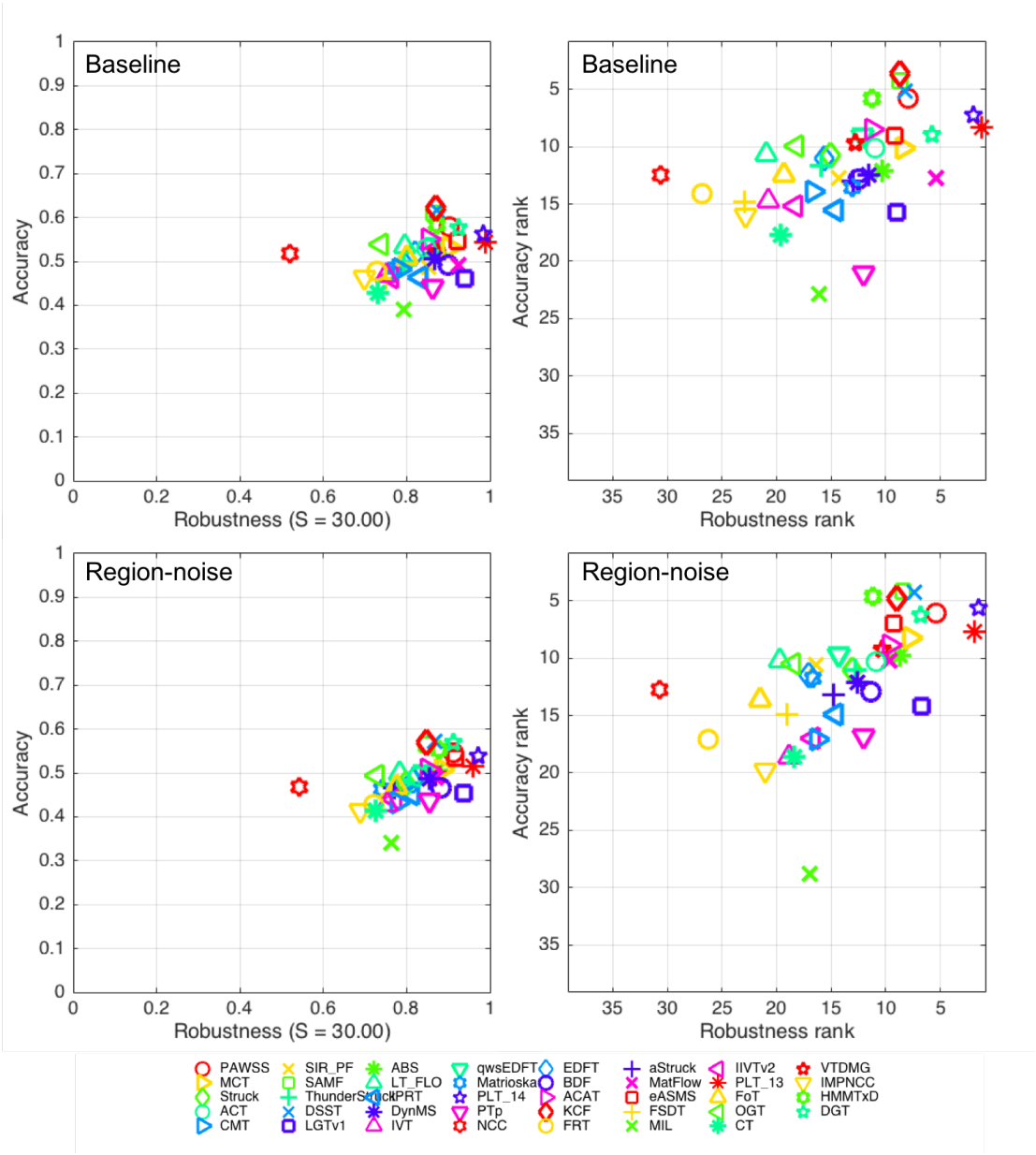


Figure 5.12: The accuracy-robustness score and ranking plots with respect to the baseline and region-noise experiments of VOT2014 dataset. Tracker is better if its result is closer to the top-right corner of the plot.

We have included the expected overlap plot in Figure 5.13. As it is shown, PAWSS is ranked fourth among all trackers, outperforming DSST and SAMF. It can be shown that the average rank is not always consistent with the expected overlap.

In [224], the authors conclude the following trackers as being either robust or very accurate: MDNet [102], DeepSRDCF [225], SRDCT [227], EBT [226], NSAMF[2], sPST [229], LDP [228], RAJSSC [230] and RobStruck[3]. We list the score/rank and expected overlap of those top trackers, the above VOT2014 top three trackers DSST [211], SAMF [212], KCF [84][4],

[2]The tracker is submitted to VOT2015 but without publication.
[3]See footnote 2.
[4]This is an improved version of the original tracker.

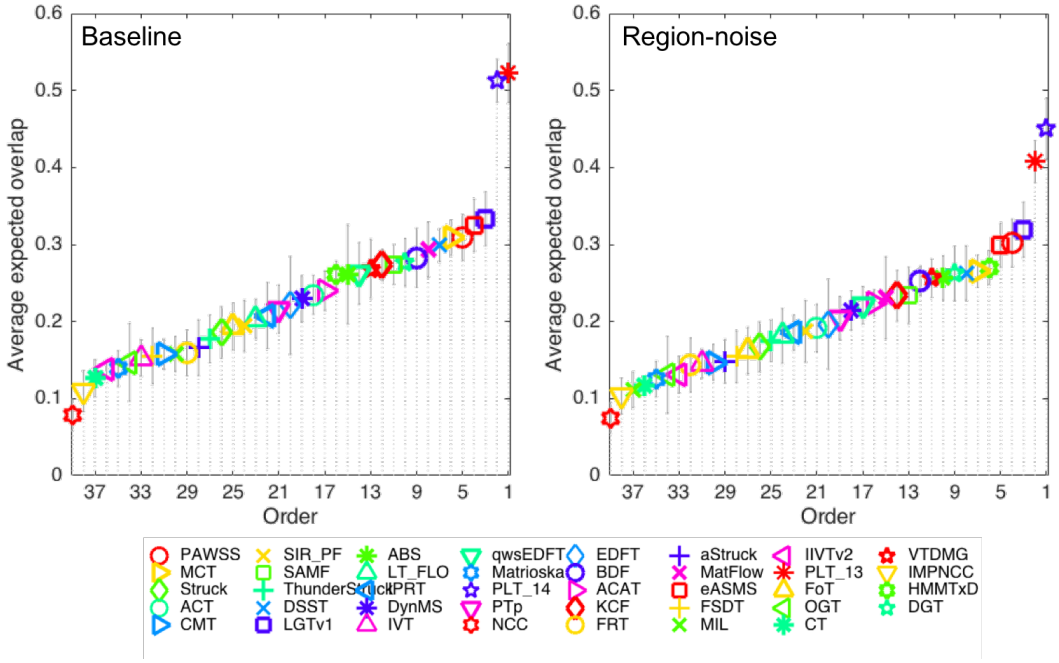


Figure 5.13: The expected overlap score ranking plots of VOT2014 dataset. Tracker is better if its result is closer to the right of the plot.

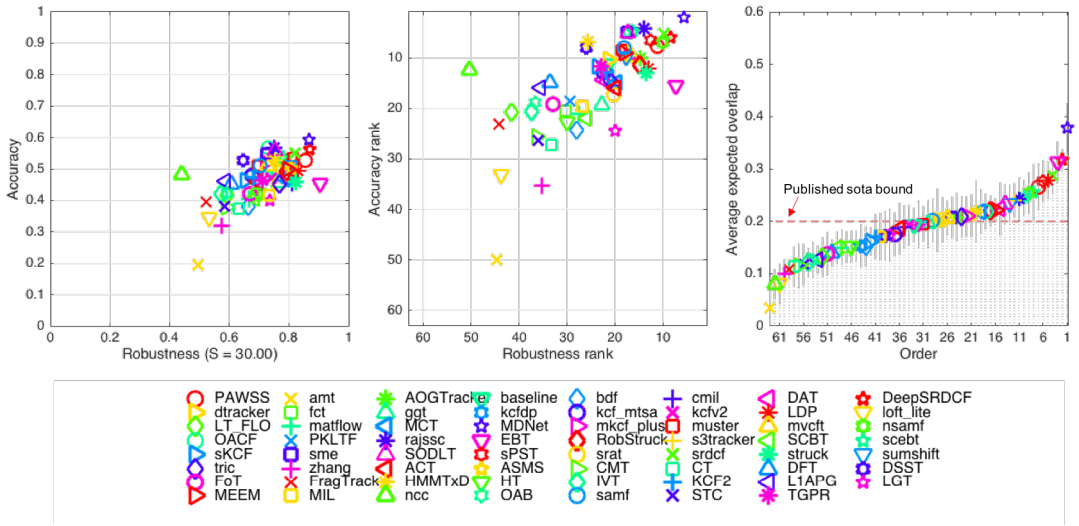


Figure 5.14: The accuracy-robustness ranking plots and the expected overlap score ranking plot of VOT2015 dataset. Tracker is better if its result is closer to the top-right corner of the plot. The published sota bound is established based on top trackers in recent years. Any tracker with performance over the boundary is considered as a state-of-the-art tracker.

plus the VOT2015 baseline NCC tracker in Table 5.7 and also shown in the expected average overlap plot Figure 5.14. According to the paper, a VOT2015 *published sota bound* criteria (0.2) is established by averaging the tracker performance published in 2014/2015 from top computer vision conferences and journals. The tracker will be considered as a state-of-the-art tracker with performance over this boundary criteria. Our tracker PAWSS is well above the criteria and is among those top trackers (ranks the 7-th, outperforming 54 trackers), also PAWSS

	Baseline				Average Rank	Expected Overlap
	Accuracy		Robustness			
	Score	Rank	Failure	Rank		
MDNet [102]	0.59	2.03	0.77	5.68	3.86	0.378
DeepSRDCF [225]	0.56	5.92	1.00	8.38	7.15	0.318
EBT [226]	0.45	15.48	0.81	7.23	11.36	0.313
SRDCT [227]	0.55	5.25	1.18	9.83	7.54	0.288
LDP [228]	0.49	12.08	1.30	13.07	12.58	0.279
sPST [229]	0.54	6.57	1.42	12.57	9.57	0.277
PAWSSb	**0.53**	**7.75**	**1.28**	**11.22**	**9.49**	**0.266**
NSAMF†	0.53	7.02	1.45	10.1	8.56	0.254
RAJSSC [230]	0.57	4.23	1.75	13.87	9.05	0.242
RobStruck†	0.49	11.45	1.58	14.82	13.14	0.220
DSST [211]	0.53	8.05	2.72	26.02	17.04	0.172
SAMF [212]	0.51	7.98	2.08	18.08	13.03	0.202
KCF [84]	0.47	12.83	2.43	21.85	17.34	0.171
NCC*	0.48	12.47	8.18	50.33	31.4	0.080

Table 5.7: VOT2015 score/ranking and expected overlap results from the top trackers of VOT2014, VOT2015 and the baseline tracker. The NCC tracker is the VOT2015 baseline tracker. Trackers marked with † are submitted to VOT2015 without publication.

achieves better than any of the VOT2014 top trackers on VOT2015 dataset.

5.5.4 Surgical Instrument Tracking

PAWSS is a general tracking framework, we also want to evaluate its performance on surgical instrument sequences. In the Endoscopic vision MICCAI2015 Challenge[5], one of the sub-challenge focuses on comparing different vision-based methods for tracking conventional and articulated instruments in laparoscopic and robotic surgery. The conventional instrument dataset is from *in vivo* laparoscopic colorectal surgeries, while the articulated instrument dataset is from *ex vivo* interventions. Both datasets are organized in the same way: they are divided into training and test data. The training data contains four 45 seconds surgery video sequences. For each instrument, the centre point of the instrument is defined as the intersection between the instrument axis and the border between the shaft and the manipulator. The annotation includes the pixel coordinates of the centre point and the normalized instrument axis vector. The test data is composed of 15 additional seconds video from each of the training sequence, and two additional new 60 second video sequences. For each instrument, the TrackedPoint of the instrument is defined and annotated as the intersection between the instrument axis and the border between the shaft and the manipulator. The dataset has not released GT for test data. The official evaluation categorized the conventional laparoscopic instrument test set according to the challenging factors including bleeding (C_{blood}), smoke (C_{smoke}), instrument occlusions ($C_{occlusion}$), multiple instruments ($C_{multiple}$) and surgical objects such as meshes and clips ($C_{objects}$). And the robotic laparoscopic instrument dataset includes sequences with multiple instruments ($C_{multiple}$). For evaluating the tracking performance, the Euclidean distance of the centre point between the GT and the tracking result of the training data is computed and compared separately for these

[5] `https://endovissub-instrument.grand-challenge.org/`

challenging factors. We submitted our proposed method to the challenge, and obtained the performance comparison from the official report.

***EndoVis* Articulated Robotic Surgical Instrument Dataset** The sequences are collected using the da Vinci® (Intuitive Surgical Inc., CA) robot with porcine tissue samples. Example frames from each sequence and annotations are shown in Figure 5.15.

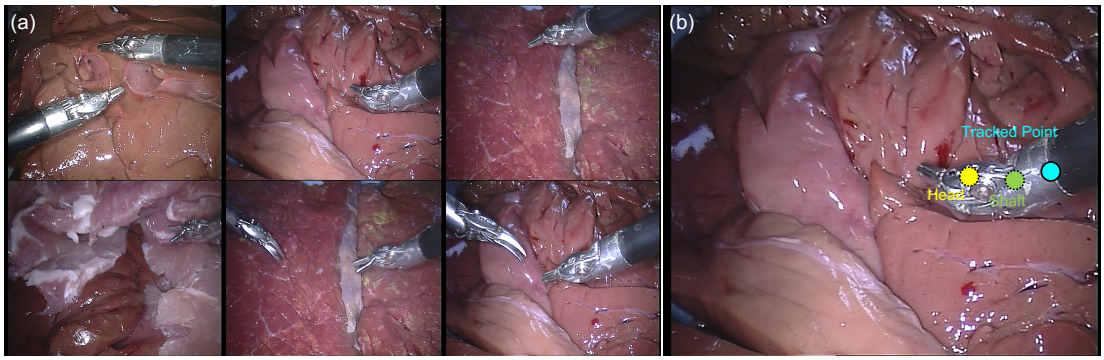


Figure 5.15: (a) Example frame from each sequence of the *EndoVis* articulated surgical instrument dataset, the last two example image is from test data; (b) The original annotation includes the position of the TrackedPoint, in our annotation, we relabeled the TrackedPoint and also added new annotations for the Head and Shaft points, which are referred as the HeadPoint and the ShaftPoint.

Original Annotation We have summarized the frame number for each sequence and have shown the accuracy evaluation separately in the original annotation section of Table 5.8 and Figure 5.17 Left. The accuracy is defined as the percentage of tracked frames within the error threshold. Distance (pixels) is averaged over correctly tracked frames. In Figure 5.17, it shows the accuracy under different threshold. In the four train sequences, there are five instruments to be tracked. The average accuracy score for the train data is 79.01% for 20 pixel threshold, with a distance error of 8.00 pixels. It is noted that the accuracy score (36.55% for 20 pixel threshold) for sequence 4 is relatively lower compared with the rest sequences. As we have summarized, the target is out of view several times in sequence 4, reaching 67 frames out of 1123 frames. Tracking-by-detection methods typically cannot handle out-of-view scenario without additional re-detection module. The underlying assumption is that the target is always in the frame view, which means whenever the target is out of frame, the tracker will gradually drift away. This explains the low accuracy of the performance, if the threshold is increased to 30 pixels, the performance has significantly improved, achieving 82.67% for accuracy. We show some tracking result examples in Figure 5.16. The TrackedPoint and bounding box are shown in cyan colour, with the GT point shown in green colour. The first column is the first frame of each sequence. As we can see, the quality of the annotation is not consistent through the whole sequence. On certain frames, the annotation is drifted and is not labelled where it is supposed to be. This would certainly affect our performance evaluation result. It is also observed that whenever the instrument is close to the frame border, the tracker will stick to the border and not track the instrument well.

High quality Annotation Since the original annotation does not provide consistent GT, the accuracy result does not reflect the true performance. We manually relabeled the TrackedPoint

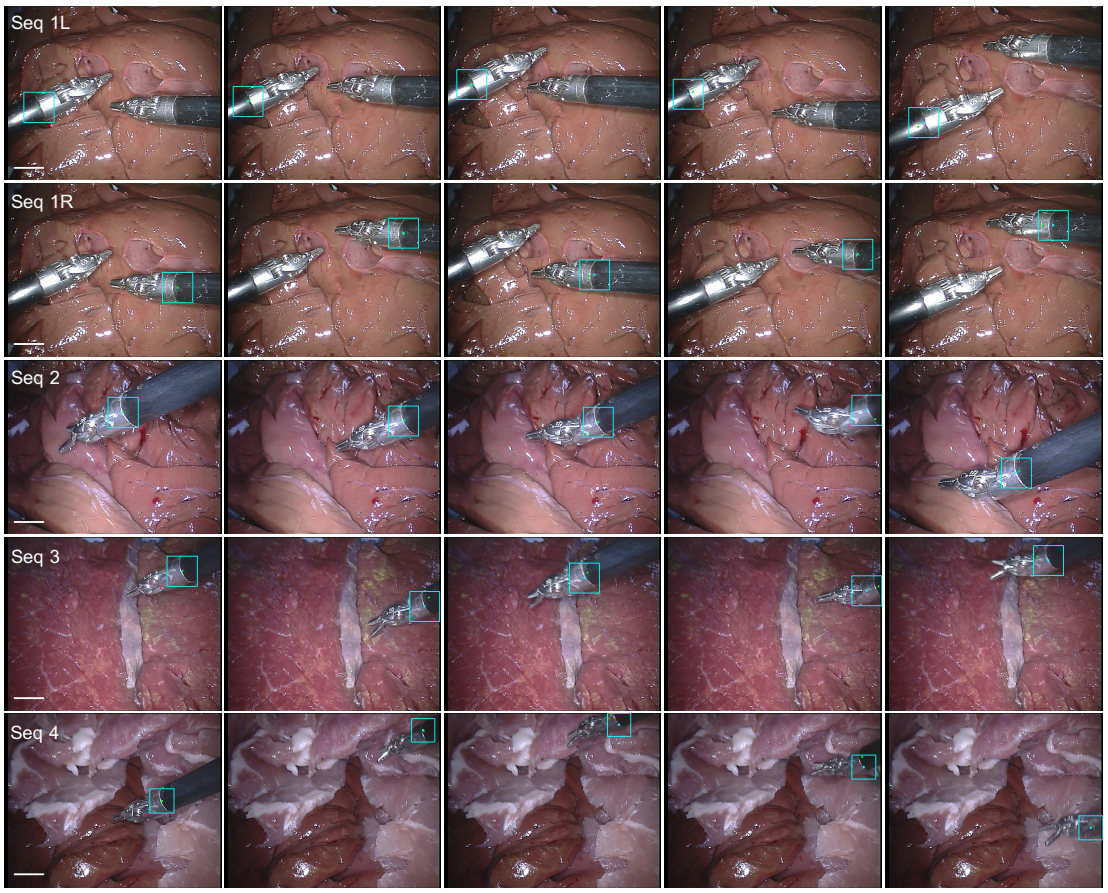


Figure 5.16: Result example frames from each sequence of the *EndoVis* articulated surgical instrument dataset. The result bounding box and centre point is represented in cyan colour, and the GT centre point is represented in green colour. Scale bar equals 100 pixels.

for the training data. Besides, we also labelled multiple joints of the instrument in the original dataset, and use the annotation for pose estimation, the details of the annotation will be discussed in Chapter 6. Here we also tracked and evaluated the HeadPoint and ShaftPoint annotations shown in Figure 5.15 (b)). The tracking performance evaluation is listed in the high quality annotation section of Table 5.8 and Figure 5.17 right. With the new annotation, our average accuracy has increased to 98.56% for 20 pixel threshold, with distance error of 6.65 pixels.

We also tracked and evaluated on the HeadPoint and ShaftPoint joints we defined in our high quality annotation. As shown in Figure 5.18. Some tracking examples are shown for the HeadPoint and ShaftPoint joint in the top and bottom row respectively. The tracking accuracy evaluation results are displayed in Table 5.9 and Figure 5.19. Our average accuracy has reached 99.96% and 99.68% for 20 pixels threshold, with distance error of 5.68 and 6.51 pixels, respectively.

In Table 5.10, the distance error (pixel) was computed and compared separately for challenging factor multiple instrument (C_{multiple}) with all the submitted methods KIT, UGA, MOD and our method PAWSS. From the official report, PAWSS outperforms all the other methods with the lowest average distance error 29.66 pixels.

	Seq 1L	Seq 1R	Seq 2	Seq 3	Seq 4	Whole
Original Annotation						
In-view (IV) and Out-of-view (OV) Frame Number						
IV	1107	1107	1096	1118	1056	5484
OV	0	0	29	6	67	102
Total	1107	1107	1125	1124	1123	5586
Accuracy ($Thres = 20$ px)						
Acc. (%)	85.00	92.86	90.60	88.10	36.55	79.01
Dist. (px)	7.42	7.07	7.41	9.64	9.26	8.00
Accuracy ($Thres = 30$ px)						
Acc. (%)	99.37	96.93	96.35	95.80	82.67	94.33
Dist. (px)	9.76	7.80	8.36	10.71	18.07	10.67
High Quality Annotation						
In-view (IV) and Out-of-view (OV) Frame Number						
IV	1107	1107	1099	1105	1066	5484
OV	0	0	26	19	57	102
Total	1107	1107	1125	1124	1123	5586
Accuracy ($Thres = 20$ px)						
Acc. (%)	100.0	99.73	98.91	98.28	95.78	98.56
Dist. (px)	4.89	9.87	3.29	4.31	11.13	6.65
Accuracy ($Thres = 30$ px)						
Acc. (%)	100.0	100.0	99.36	99.46	99.72	99.71
Dist. (px)	4.89	9.90	3.38	4.56	11.57	6.83

Table 5.8: Accuracy of *EndoVis* Articulated Robotic Surgical Instrument Train Data for the Tracked-Point

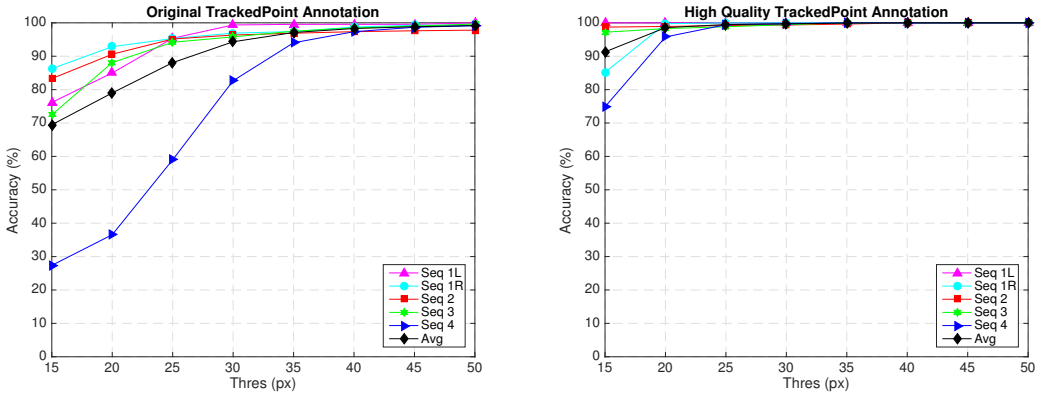


Figure 5.17: Tracking accuracy of *EndoVis* Articulated Robotic Surgical Instrument training data under different accuracy threshold with the original and high-quality annotations

***EndoVis* Conventional Laparoscopic Instrument Dataset** Compared to the *ex vivo* robotic instrument dataset, the conventional instrument sequences reflect complex challenges during surgery, including smoke, bleeding, blurry and various kinds of instruments. In Table 5.11, the distance error (pixel) was computed and compared separately for each challenging factor with all the submitted methods KIT, UGA and our method PAWSS. From the official report, PAWSS

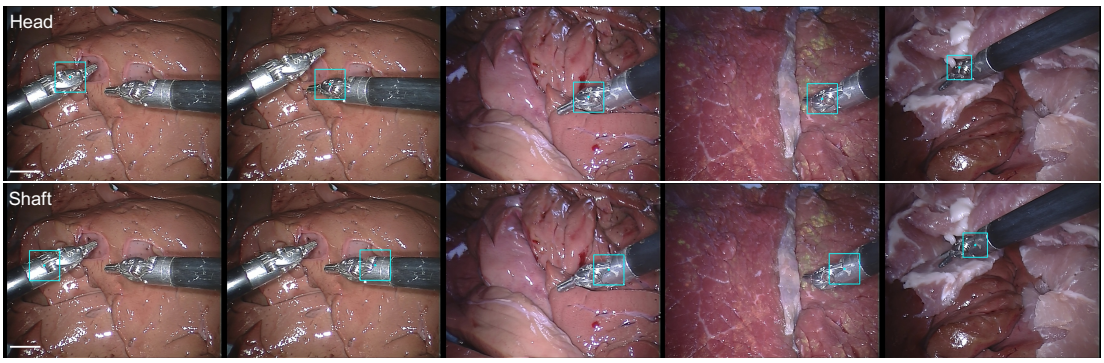


Figure 5.18: Result example frames from each sequence of the *EndoVis* articulated surgical instrument dataset for HeadPoint joint (top row) and ShaftPoint joint (bottom row). The result bounding box and centre point is represented in cyan colour, and the GT centre point is represented in green colour.Scale bar equals 100 pixels.

	Seq 1L	**Seq 1R**	**Seq 2**	**Seq 3**	**Seq 4**	**Whole**
In-view (IV) and Out-of-view (OV) Frame Number						
IV	1107	1107	1125	1124	1123	5586
OV	0	0	0	0	0	0
Total	1107	1107	1125	1124	1123	5586
HeadPoint Accuracy ($Thres = 20$ px)						
Acc. (%)	100.0	100.0	99.82	100.0	100.0	99.96
Dist. (px)	3.06	4.10	10.32	4.52	6.33	5.68
ShaftPoint Accuracy ($Thres = 20$ px)						
Acc. (%)	100.0	98.46	100	99.91	100	99.68
Dist. (px)	2.48	12.08	6.82	4.79	6.48	6.51

Table 5.9: Accuracy of *EndoVis* Articulated Robotic Surgical Instrument Train Data for HeadPoint and ShaftPoint joints with High Quality Annotation

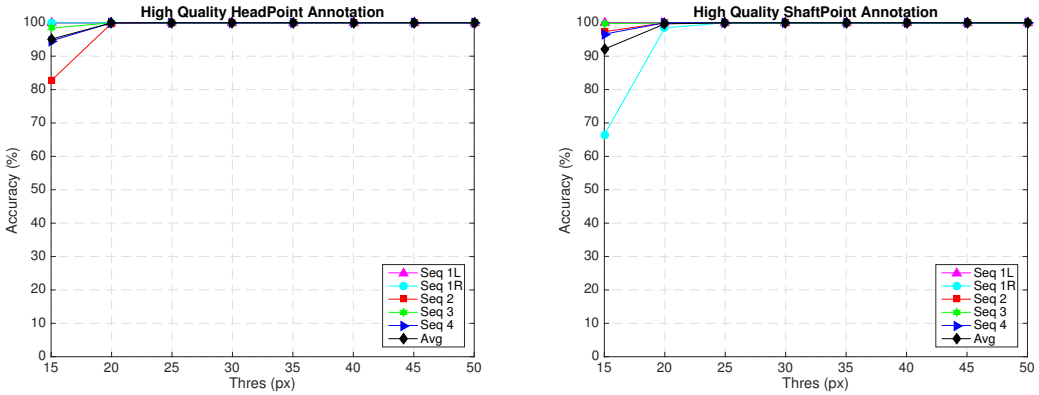


Figure 5.19: Accuracy of *EndoVis* Articulated Robotic Surgical Instrument training data under different accuracy threshold with high quality annotation

outperforms all the other methods in every challenging subset with the lowest average distance error 96.78 pixels. We show some tracking result examples in Figure 5.21. The TrackedPoint is shown in cyan colour, and the first column is the first frame of each sequence in the test set.

GHT Surgical Instrument Experiments We tested on the *ex vivo* instrument dataset in

	$C_{multiple}$	Whole
KIT	113.91	106.60
UGA	40.73	34.94
MOD	45.12	40.16
PAWSS	**38.36**	**29.66**

Table 5.10: Distance (pixel) comparison with all the submitted methods for the TrackedPoint of the robotic laparoscopic instrument test set. Multiple instrument challenging subset is evaluated separately.

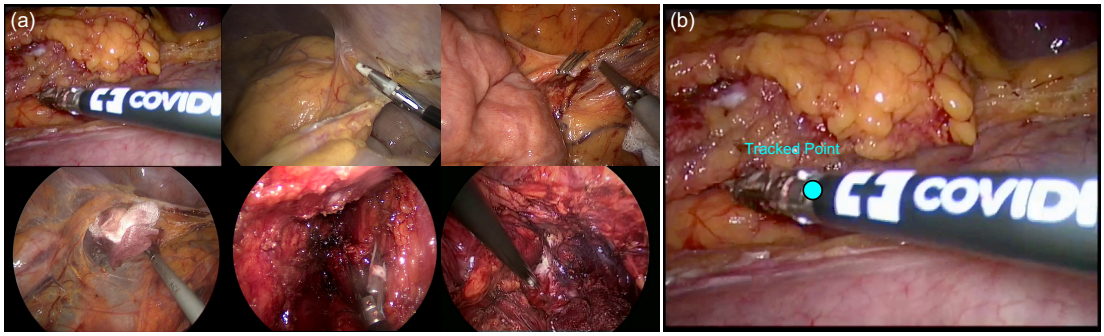


Figure 5.20: (a) Example frame from each sequence of the *EndoVis* articulated surgical instrument training dataset; (b) The annotation includes the position of the TrackedPoint.

	C_{blood}	$C_{multiple}$	$C_{objects}$	$C_{occlusion}$	C_{smoke}	Whole
KIT	233.62	220.87	117.23	225.58	193.85	178.89
UGA	276.44	235.42	228.04	193.82	231.87	217.91
PAWSS	**181.59**	**110.85**	**68.29**	**87.11**	**96.31**	**96.78**

Table 5.11: Distance (pixel) comparison with all the submitted methods for the TrackedPoint of the conventional laparoscopic instrument test set. Various challenging subsets are evaluated separately.

Chapter 4 to compare the performance of GHT and PAWSS, along with the CST [6] and TLD [3] trackers. The centre trajectory of the target, precision plot and the box plot results are shown in Figure 5.22. All the plots follow the same metrics in Section 4.9.1. It is shown that for most sequences, PAWSS achieves the best performance among all the methods, except for Dataset I, in which the instrument is occluded by tissue samples. However, even the above results show that PAWSS works well in Dataset II and III, it is worth noting that like most tracking-by-detection tracking methods, PAWSS is not designed with a re-detection component. Whenever the target is occluded for a long time, the tracker will potentially drift away from the target area. While the result of Dataset IV demonstrates that PAWSS is capable of long term robust tracking.

***In Vivo* Surgical Instrument Experiments** We also test on some other *in vivo* sequences and show the result in Figure 5.23. As we can see, the tracker works well even under complex *in vivo* environment.

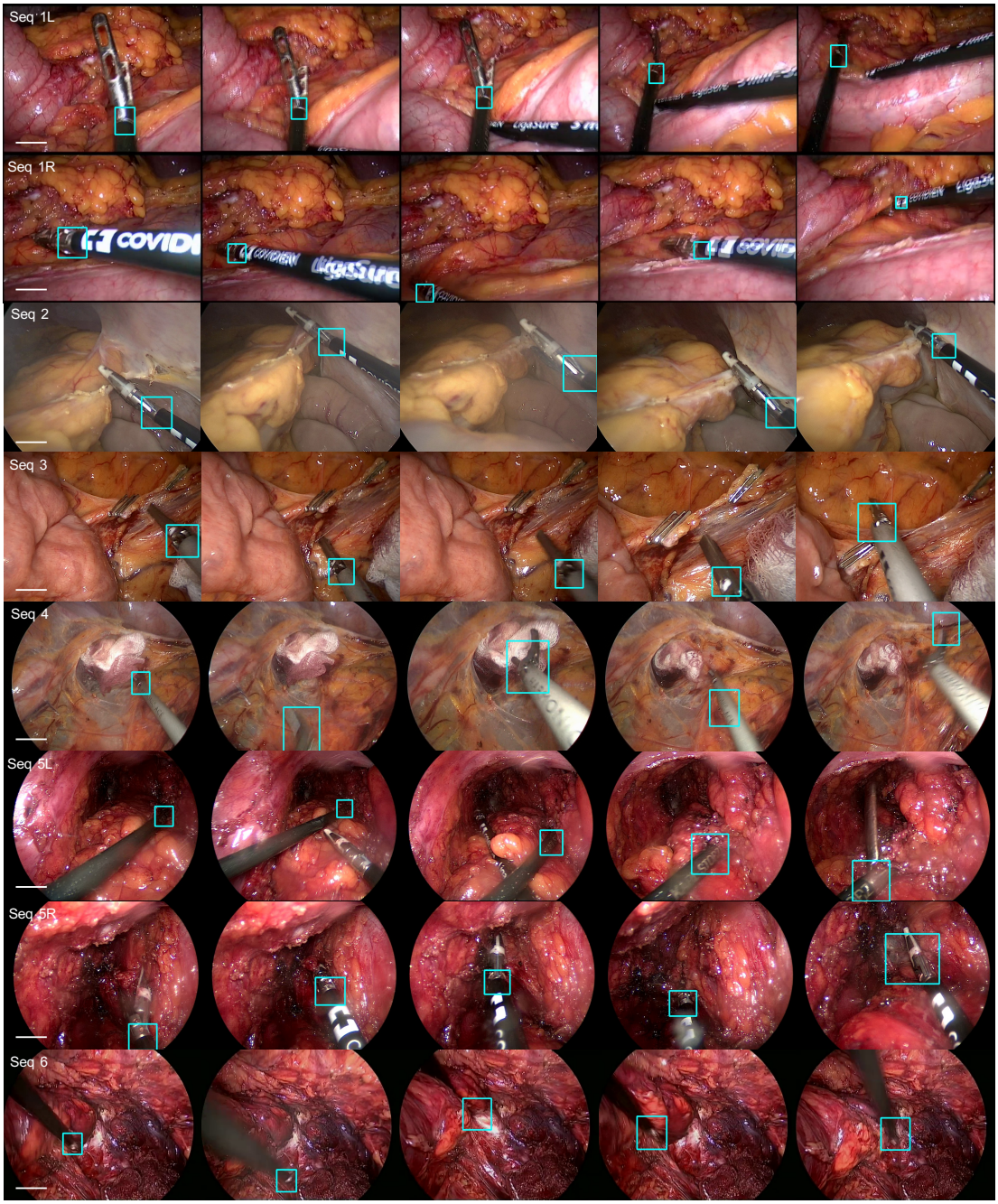


Figure 5.21: Result example frames from each test sequence of the *EndoVis* conventional surgical instrument dataset. The result bounding box is represented in cyan colour. Scale bar equals 100 pixels.

5.6 Discussion

In this chapter, we propose a tracking-by-detection framework, called PAWSS, for online object tracking. Corrupted samples, which includes partial object or background information confuse classifier ultimately lead to tracking drift or failure.

The performance of our tracker is thoroughly evaluated on the OTB, VOT2014 and VOT2015 datasets, and is compared with recent state-of-the-art trackers. Results demonstrate that PAWSS achieves the best performance in both PR and SR in the OPE for OTB dataset. It outperforms Struck by 36.7% and 36.9% in PR/SR scores. Also, it provides a comparable PR

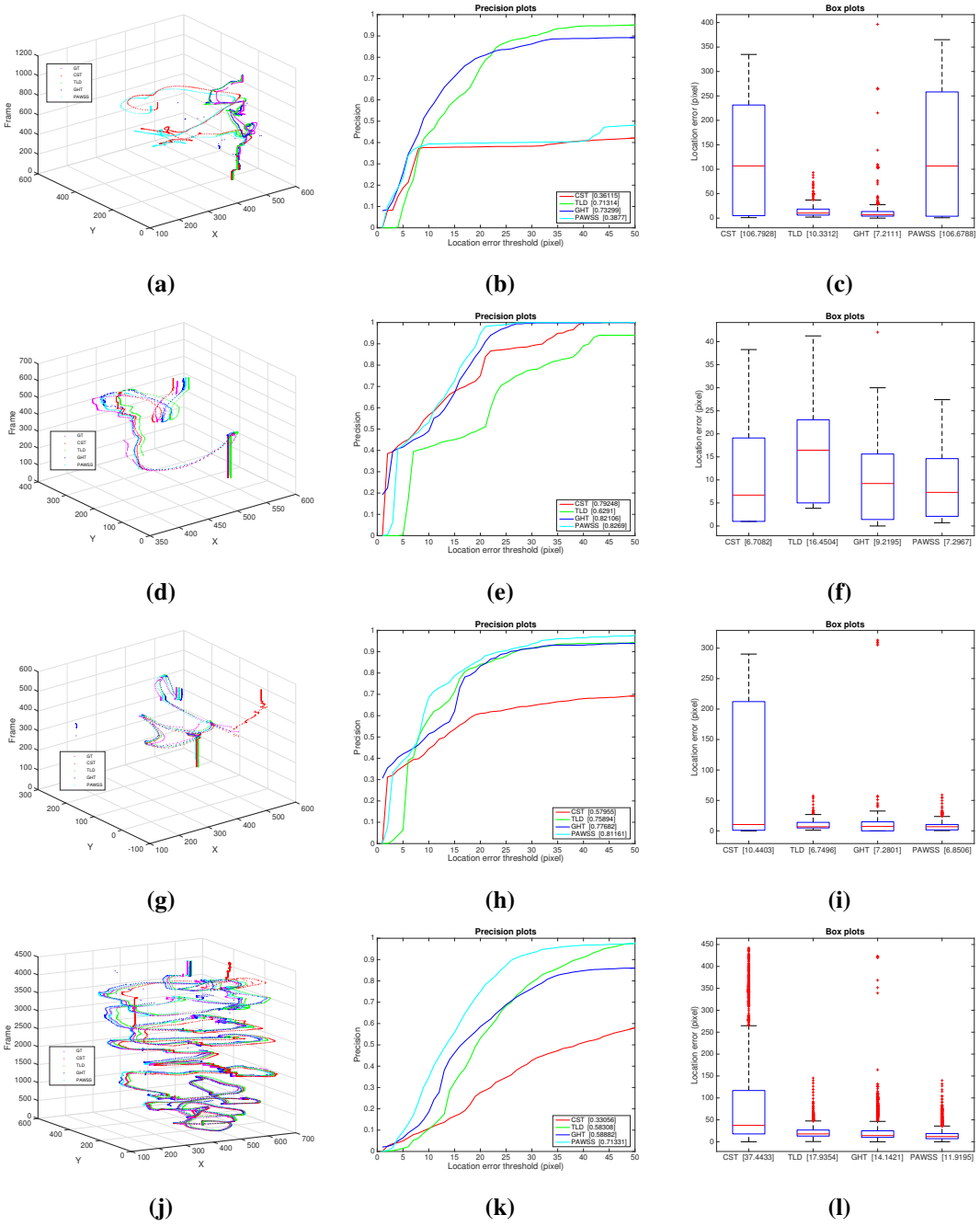


Figure 5.22: Performance comparison of our proposed tracker PAWSS with GHT and two trackers: CST [6] and TLD [3] on (a-c) Dataset I with tissue occlusion, (d-f) Dataset II with instrument occlusion, (g-i) Dataset III with out-of-view occlusion and (j-l) the extended tracking sequence Dataset IV.

score, and improves SR score by 4.8% over SOWP. On VOT2014 dataset, PAWSS has relatively lower accuracies but the lowest failure rate among the top trackers, we evaluated without re-initialization, and achieves the highest accuracies. Also on VOT2015 dataset, PAWSS is considered state-of-the-art and is among the top trackers. Compared to other tracking-by-detection trackers, the highlights of our proposed framework PAWSS can be summarised as follows

- An effective colour-based segmentation model is incorporated to assign weights to the patch-based descriptor: unlike in [2, 199], our patch weighting method is simple and

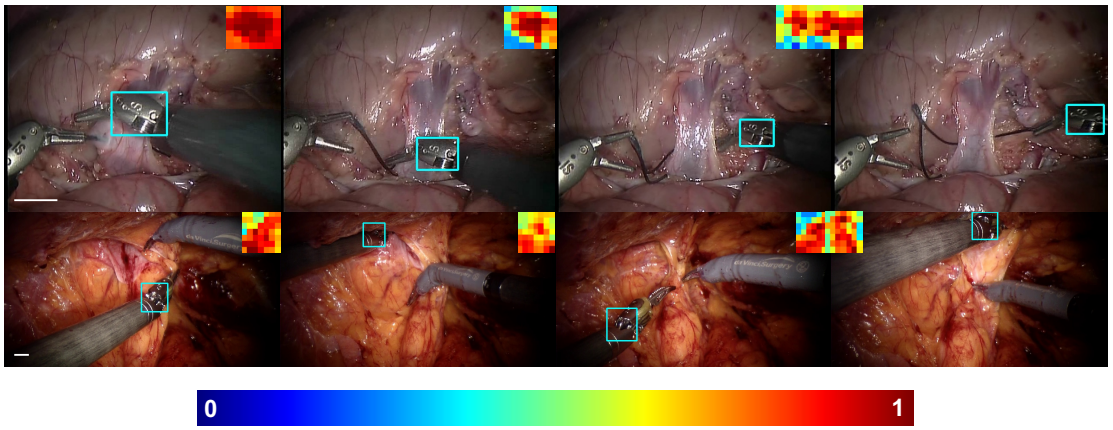


Figure 5.23: Instrument Tracking result with patch weight displayed in the top corner of the image. Scale bar equals 100 pixels.

straightforward, weight update for each patch is independent, and is only determined by the foreground and background histogram distributions from the segmentation model.

- A two-level sampling strategy: rather than training with fixed-scale samples, multi-scale samples are extracted to enrich the training pool, allowing the tracker to handle both incremental and abrupt scale variations between frames.

- Near real-time performance without any specific optimisation.

For instrument tracking, we also qualitatively and quantitatively evaluated our tracker on the public *EndoVis* Surgical Instrument Dataset and *in vivo* surgical instrument sequences. GT for the challenge test data is not available, so we submitted our tracking result to the organizers of the challenge. Since our method does not rely on any pre or offline training, we then compared our results with the official GT for the TrackedPoint in the training data, and the tracking accuracy reached 79.01% with 20 pixel threshold. Through visualization, we have shown the official annotation is not quality consistent, so we manually created a high quality multi-joint annotation for the dataset. We evaluated multiple joints (TrackedPoint, HeadPoint and ShaftPoint) on the dataset, and our performance accuracy increased to over 98% for all the joints with 20 pixel threshold. From the official report, our method has shown its excellent tracking ability on the challenge data, and also with *in vivo* sequences in complicated surgical environment.

PAWSS is based on online learning techniques without relying on any prior knowledge of the tracked target. This enables our method to be applied to general tracking task. We would also like to discuss the limitation of our tracker. First, the target location is represented by rectangle bounding box in our tracker. Even with the assistance of the segmentation model to distinguish the foreground and the background, the assumption is that the target occupies most area of the bounding box. If the target only occupies small fraction, the classifier would be polluted and misled by the background information and can easily cause tracking failure. Second, our tracker can reach semi real time for single object tracking, but it slows down linearly with more target number. Also, our tracker is designed for single object tracking for now, if multiple targets are selected, for example, when multiple joints of the instrument are to be tracked, each

target is associated with one tracker and is tracked individually without any collaboration with each other. In the future, we would like to explore more with multiple target tracking.

Chapter 6

Deep Learning Based 2D Pose Estimation for Articulated Surgical Instruments

6.1 Introduction

Robotic surgery systems such as the da Vinci® (Intuitive Surgical Inc, CA) have introduced a powerful platform for articulated instrument control in MIS through tele-operation of the surgical camera and specialised dexterous instruments. The next generation of such platforms is likely to incorporate a more significant component of computer-assisted system support through software, visualisation and analytical tools to better understand the surgical process and progress. Real-time knowledge of the instruments pose with respect to anatomical structures and the viewing coordinate frame is a crucial piece of information for such systems focused on providing assistive or autonomous surgical capabilities. The location and orientation of the instruments in the camera's frame is a crucial piece of information for such systems for advanced assistive surgical capabilities [28]. Control systems which can supply automated visual servoing [184], soft motion constraints [185] and tactile feedback [186] are reliant on knowing positional information about both the shaft and the tip of the articulated instrument. Hardware based solutions such as optical tracking systems using fiducial markers [187] require modification to the instrument design posing ergonomic challenges and additionally suffer from robustness issues due to line-of-sight requirements. In principle, direct use of robotic joint encoders and forward kinematics to track instruments is possible in robot-assisted interventions. However, in the da Vinci®, the kinematic chain involves 18 joints, which is more than 2 meters long. This is challenging for accurate absolute position sensing and requires time-consuming hand-eye calibration between the camera and the robot coordinates. On cable driven systems the absolute error can be up to 1 inch, which means the positional accuracy is potentially too low for tracking applications without visual correction [139, 184, 188]. Pure image based solutions [137, 189, 190] directly estimate the instrument pose in the reference frame of the observing camera. This avoids complex calibration routines and can be implemented entirely through software which allows them to be applied retrospectively and without modification to the instruments or the surgical workflow. While most of these methods have focused on semantic segmentation of the image or on single landmark detection on the instrument tip, which cannot represent the full pose of an instrument or include articulation. Additional challenges to articulated tracking in surgical video are because information inferred from video directly can

suffer from occlusions, noise and specularities, perspective changes and bleeding or smoke in the scene.

Image-based surgical instrument tracking and pose estimation has been shown to be feasible in different specialisations, such as retinal microsurgery [150], neurosurgery [192] and MIS [137, 231]. While detection and tracking are difficult, pose estimation presents additional challenges due to the complex articulation structure. Most image-based methods [150, 231] often extract low-level visual features from keypoints or regions to learn offline or online part appearance templates by using traditional machine learning algorithms. They predominantly estimated the instrument pose in 2D by estimating image based translation parameters, scale and in-plane rotation without explicitly modelling the 3D shape of the instrument. These have been based around low-level image processing [191] which accumulate hand-crafted visual features and more complex learned discriminative models [192, 190] which track an instrument by performing detection independently on each frame. Such methods are typically fast and robust, handling complex and fast motion as well as recovery when the instrument is occluded by the field of view of the camera or smoke and tissue as they perform a global or semi-global search of the entire image for the tracked instrument. Fewer methods have attempted to estimate the 3D pose of the instruments directly from image data. This typically is a much more complex problem as it involves estimating three additional DOF from very weak small baseline stereo or monocular cues. However, it provides additional benefits over 2D methods as it allows reasoning about instrument-instrument occlusions and interactions with tissue surfaces. Most of these methods focus on the alignment of a 3D model with a probabilistic classification of the image [135, 136, 134] which allows the fusion of geometric constraints with image data without an offline learning phase. A significant challenge with 3D tracking methods is that they commonly fail when the instrument motion is fast or complex, as they restrict the parameter search to local regions close the the estimated parameters from the previous frame. In many cases this can lead to drift which requires a manual reset of the tracking.

Such low-level feature representations usually suffer from a lack of semantic interpretation, which means they cannot capture the high level domain appearance. To improve robustness it is possible to integrate external constraints such as surgical Computer-aided Design models [137, 232] or robotic kinematics [233, 139] but the essential image-driven approach is still central to providing robust and generalisable systems. Deep convolutional networks have emerged as the method of choice for various visual tasks [97, 98, 90, 234]. They compose multiple layers of simple but non-linear modules into a higher and abstract representation. The key aspect of deep learning is that these layers of features are not designed by human engineers: they are learned from massive data using a general-purpose learning procedure [235]. This methodology has been applied to medical images [236] and datasets to drive deep systems have begun emerging for recognition tasks in laparoscopic videos [140]. The methodology has been demonstrated to be effective in instrument presence detection [141]. Additionally, networks for semantic instrument segmentation have also been proposed and shown to be effective in real-time performance [237]. However, few methods are yet able to jointly detect the instrument contour and to estimate articulation from it.

6.2 Model Architecture

Following the deep learning paradigm, we present a novel 2D pose estimation framework for articulated endoscopic surgical instruments, which involves a detection-regression fully convolutional network (FCN) and a multi-instrument parsing component. To achieve the articulation performance we seek, we re-annotated instrument joints of the dataset presented at the *EndoVis Challenge*, MICCAI 2015 and used this for training our network. Our method achieves very compelling performance and illustrates some interesting capabilities including transfer between different instrument sets and between *ex vivo EndoVis* and *in vivo* data. The high-level of detail annotations which we have created as part of this study will naturally be made available for future research (See Figure 6.7).

The overall pipeline of our CNN-based framework is shown in Figure 6.1. In this section, we first define the instrument joint structure. Then we introduce the objective and architectural design of each module of our detection-regression FCN. In our detection-regression architecture, the detection module guides the subsequent regression module to focus on the joint parts, and the regression module helps the detection module to localize joints more precisely. Finally, we describe how the network output is integrated for inferring the poses of multiple instruments.

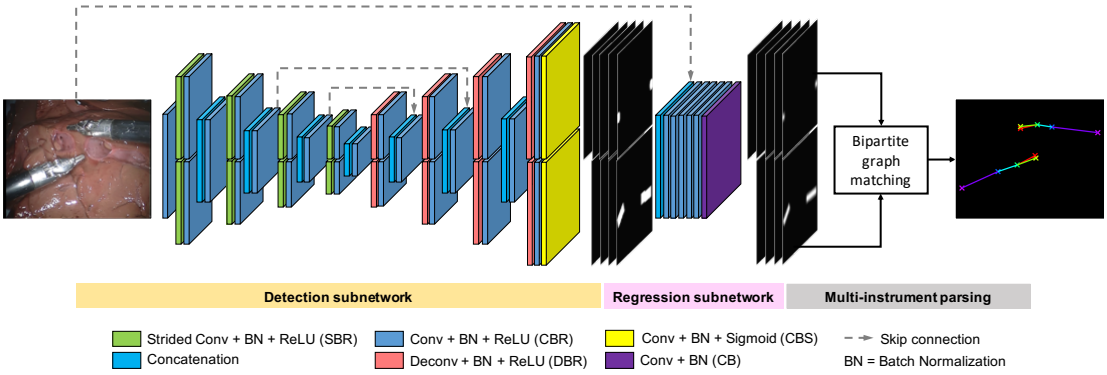


Figure 6.1: The pipeline of our proposed pose estimation framework and the detection-regression FCN architectural design. The output of the network is integrated to associate joints and assemble them into the final poses for all instruments in the frame.

The pose of an articulated instrument can be represented in different ways. For example, it can take advantage of kinematic information by using joint relative orientation. In our work, we rely on pure visual cues, an articulated instrument is decomposed as a tree structure of individual joint parts as seen in Figure 6.2, a joint pair is defined as a pair of joints which are connected according to the skeleton. Based on the articulation, instruments in different datasets are represented with a similar tree structure which is made up of N joints and M joint pairs. Therefore, instrument pose estimation task can be reduced to detecting the location of individual joint parts, and if there are multiple instruments present in the image, joints of the same instrument should be correctly associated after detection. Our bi-branch model architecture is inspired by CMUPose [98]. Joint locations and associations between joint pairs are learnt jointly via two branches of same encoder-decoder predication process. In each of the blocks, features or predictions from each branch capture different structural information about the instrument and are

concatenated for the next block.

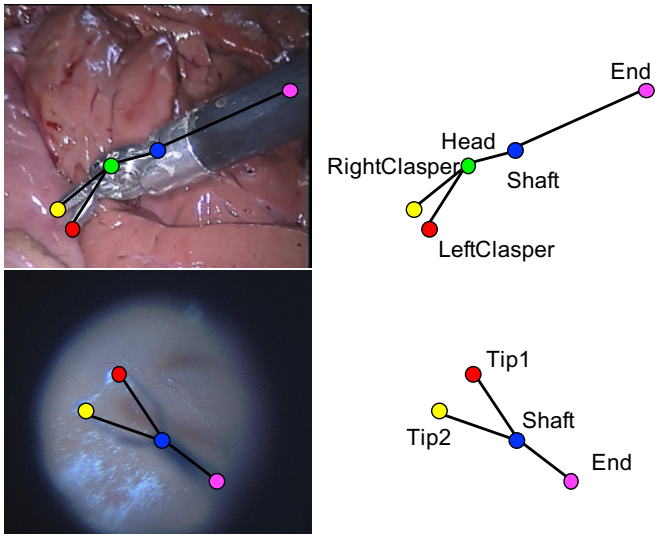


Figure 6.2: The instrument structure is decomposed into N joints and M joint pairs, based on the articulation, instruments for different datasets could have slightly different joint structure. Joints are represented by colour dots, and joint pairs are connected by black lines. (Top) The *EndoWrist* instrument is made up of 5 joints and 4 joint pairs; (Bottom) The Retinal instrument is made up of 4 joints and 3 joint pairs.

6.3 Joint Detection and Association Subnetwork

Inspired by the recent success of FCNs [98, 236], we design our bi-branch joint detection and association network. To train our network, since joints could overlap with each other, the detection task is treated as a set of binary-class problems, instead of a multi-class problem.

In our bi-branch network, the first branch is used to predict N individual joint probability maps, one for each joint; and the second branch is used to predict the M joint association probability maps, one for each joint pair. Therefore, the ground truth for the detection subnetwork is constructed as a set of $N + M$ binary maps. We used the popular downsampling-upsampling FCN architecture. The FCN encoder-decoder network architecture concept is widely used for semantic segmentation problems since it transfers from classification to dense pixel-wise prediction probability maps with the same size as the input image. Compared to the FCN architecture, the earlier patch-based classification deep learning approaches [238], where each pixel is classified using a patch of image around it usually employs fully connected layers. Those fully connected layers limit the size of the input patches to be fixed. In the FCN architecture, fully connected layers are turned into convolution layers, which has the advantages such as reduced number of parameters, faster forward-backward pass speed or taking images of arbitrary sizes [234]. We also augmented our model with skip connections by fusing features from different layers to refine the spatial output precision. We take the Shaft-End joint pair as example, and illustrate the corresponding ground truth in Figure 6.3. For joint ground truth map (Figure 6.3 (c-d)), the pixels located within a certain radius r_d of the labelled location are considered as the joint, and are set to 1, and the remaining pixels are considered as background, and are set to 0. To reflect the connection relationship and to measure the association of correct joints, the association ground map is constructed as shown in Figure 6.3 (b). The pixels within distance

r_d to the line connecting the joints are set to foreground, which form a rotated rectangle and are set to 1, other pixels are considered as background and are set to 0. The specifications of

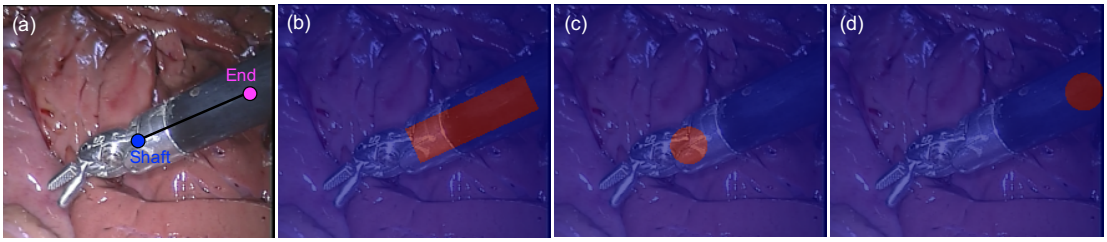


Figure 6.3: Detection subnetwork GT example for Shaft-End joint pair (a): the binary map for Shaft-End pair association map (b), the Shaft (c) and End (d) joint.

	Kernel (Size, Stride)	**Output (Channel× H×W)**
	Downsample	
CBR	$3 \times 3, 1 \times 1$	$64 \times h \times w$
Branch SBR1	$2 \times 2, 2 \times 2$	$64 \times h/2 \times w/2$
Branch CBR1	$3 \times 3, 1 \times 1$	$64 \times h/2 \times w/2$
CBR1	$1 \times 1, 1 \times 1$	$128 \times h/2 \times w/2$
Branch SBR2	$2 \times 2, 2 \times 2$	$128 \times h/4 \times w/4$
Branch CBR2	$3 \times 3, 1 \times 1$	$128 \times h/4 \times w/4$
CBR2	$1 \times 1, 1 \times 1$	$256 \times h/4 \times w/4$
Branch SBR3	$2 \times 2, 2 \times 2$	$256 \times h/8 \times w/8$
Branch CBR3	$3 \times 3, 1 \times 1$	$256 \times h/8 \times w/8$
CBR3	$1 \times 1, 1 \times 1$	$512 \times h/8 \times w/8$
Branch SBR4	$2 \times 2, 2 \times 2$	$512 \times h/16 \times w/16$
Branch CBR4	$3 \times 3, 1 \times 1$	$512 \times h/16 \times w/16$
CBR4	$1 \times 1, 1 \times 1$	$1024 \times h/16 \times w/16$
	Upsample	
Branch DBR1	$2 \times 2, 2 \times 2$	$256 \times h/8 \times w/8$
Branch CBR1	$3 \times 3, 1 \times 1$	$256 \times h/8 \times w/8$
CBR1	$1 \times 1, 1 \times 1$	$512 \times h/8 \times w/8$
Branch DBR2	$2 \times 2, 2 \times 2$	$128 \times h/4 \times w/4$
Branch CBR2	$3 \times 3, 1 \times 1$	$128 \times h/4 \times w/4$
CBR2	$1 \times 1, 1 \times 1$	$256 \times h/4 \times w/4$
Branch DBR3	$2 \times 2, 2 \times 2$	$64 \times h/2 \times w/2$
Branch CBR3	$3 \times 3, 1 \times 1$	$64 \times h/2 \times w/2$
CBR3	$1 \times 1, 1 \times 1$	$128 \times h/2 \times w/2$
Branch DBR4	$2 \times 2, 2 \times 2$	$32 \times h \times w$
Branch CBR4	$3 \times 3, 1 \times 1$	$32 \times h \times w$
CBR4	$1 \times 1, 1 \times 1$	$64 \times h \times w$
CBS	$1 \times 1, 1 \times 1$	$(M + N) \times h \times w$

Table 6.1: The Network Specifications for the Detection Subnetwork: The Kernel Size and Stride, and the Output Size (Channel × Height × Width) of Each Layer. The Original Dimension of the Input Image is $3 \times h \times w$, and the Network Outputs stacked $(M + N)$ Probability Maps with the Same Size as the Input Image.

the network are shown in Table 6.1. We followed the U-Net [236] architecture. As shown in Figure 6.1, high level encoder features are concatenated with the upsampled decoder output.

And deconvolution is followed by convolution layers, which learn to assemble a more precise output based on the fused features. Instead of pooling operations, we use strided convolution for downsampling and also eliminate fully connected layers and use all convolutional layers following the recent examples from the literature [234]. Larger kernels usually contain more parameters, and are computationally expensive, so to keep both number of parameters and the amount of computation contained, we employ small convolution kernels.

It is trained with a per-pixel binary cross-entropy loss function L_d which is defined as:

$$L_d = \frac{1}{(M+N)\Omega} \sum_{k=1}^{M+N} \sum_{\mathbf{x}\in\Omega} \left[p_{\mathbf{x}}^k \log \tilde{p_{\mathbf{x}}^k} + \left(1 - p_{\mathbf{x}}^k\right) \log \left(1 - \tilde{p_{\mathbf{x}}^k}\right) \right] \tag{6.1}$$

where $p_{\mathbf{x}}^k$ and $\tilde{p_{\mathbf{x}}^k}$ denotes the k-th GT and the corresponding sigmoid output at pixel location $\mathbf{x}$ in the frame domain Ω.

6.4 Regression Subnetwork

From the pixel-wise prediction output of the detection network, we could obtain coarse location of each joints, but in order to obtain precise location of the joints, we add a regression network (see Figure 6.1) following the detection network.

The input of the network is the concatenation of the input image and the stacked $M + N$ output probability maps of the detection network, with the latter acting as a semantic guidance for the regression network to focus on the joint parts and their structural relationships. Previous work [97] showed that directly regressing single points from an input frame is highly non-linear, so instead of regressing single points, the network will produce stacked joint density maps, which have the same size as the input image. The network contains 5 *Conv+Batch Normalization+ReLU* blocks, followed by a *Conv+Batch Normalization* block. The specifications of the network is shown in Table 6.2.

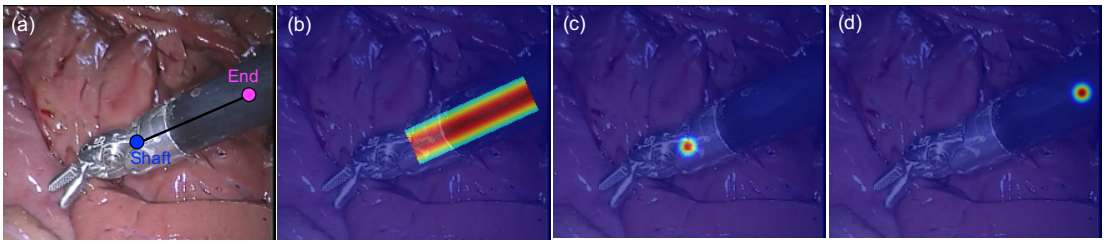


Figure 6.4: Detection subnetwork GT example for Shaft-End joint pair (a): the binary map for Shaft-End pair association map (b), the Shaft (c) and End (d) joint.

In Figure 6.4, we illustrate the Shaft-End joint pair ground truth maps for the regression subnetwork. For joint ground truth maps (Figure 6.4 (c-d)), each joint annotation corresponds to an density map which is formed with a $2D$ Gaussian centred at the labelled point location. And the association ground truth density maps are represented with a Gaussian distribution along the joint pair centre line, with a standard deviation σ shown in Figure 6.4 (b).

Therefore, the goal of the regression subnetwork is to regress the density maps from the input image with the guidance of the detection probability maps. It is trained with the mean

	Kernel (Size, Stride)	Output (Channel× W×H)
CBR1	$3 \times 3, 1 \times 1$	$64 \times w \times h$
CBR2	$3 \times 3, 1 \times 1$	$128 \times w \times h$
CBR3	$3 \times 3, 1 \times 1$	$256 \times w \times h$
CBR4	$3 \times 3, 1 \times 1$	$256 \times w \times h$
CBR5	$1 \times 1, 1 \times 1$	$256 \times w \times h$
CB	$1 \times 1, 1 \times 1$	$(M+N) \times w \times h$

Table 6.2: The Network Specifications for Regression Subnetwork: The Kernel Size and Stride, and the Output Size (Channel × Height × Width) of Each Layer. The Regression Network is Fed with the Concatenation of the Input Image and the Detection Output Maps, and Outputs stacked $(M+N)$ Probability Maps with the Same Size as the Input Image.

squared loss L_r which we define as:

$$L_r = \frac{1}{(M+N)\Omega} \sum_{k=1}^{M+N} \sum_{\mathbf{x} \in \Omega} \left\| h_{\mathbf{x}}^k - \tilde{h_{\mathbf{x}}^k} \right\|^2 \tag{6.2}$$

where $h_{\mathbf{x}}^k$ and $\tilde{h_{\mathbf{x}}^k}$ represent the predicted and the kth GT heatmaps at pixel location $\mathbf{x} \in \Omega$, respectively.

6.5 Multi-instrument Parsing

After obtaining the heatmaps of all the joints and associations, non-maximum suppression (NMS) [239] is performed on the joint heatmaps to get joint candidates D_N from potential multiple instruments. NMS is popularly used in deep learning and generally in computer vision to eliminate redundant candidates. It selects high-scoring candidate and skips ones that are close to an already selected candidate.

$$D_N = \left\{ \mathbf{d}_n^k \mid n \in \{1 \dots N\},\ k \in \{1 \dots K_n\} \right\} \tag{6.3}$$

where N is the number of joint types of the instrument structure, and K_n is the number of candidates of the n-th joint type, and $\mathbf{d}_n^k \in \mathbb{R}^2$ is the 2D location of the k-th candidate of the n-th joint type. To associate all the joints belonging to the same instrument, we define an indicator $c_{n_1 n_2}^{k_1 k_2} \in \{0, 1\}$ to show if two joint candidates $\mathbf{d}_{n_1}^{k_1}$ and $\mathbf{d}_{n_2}^{k_2}$ are connected or not. The final goal is to find an optimal matching C for all the possible connection pairs:

$$C = \left\{ c_{n_1 n_2}^{k_1 k_2} \mid n_1, n_2 \in \{1 \dots N\},\ n_1 \neq n_2,\ k_1 \in \{1 \dots K_{n_1}\},\ k_2 \in \{1 \dots K_{n_2}\} \right\} \tag{6.4}$$

As shown in Figure 6.5, circles with different colour represent the detected joint candidates $D_n = \{\mathbf{d}_n^k \mid k \in \{1 \dots K_n\}\}$ of different joint type. Instead of a fully connected graph (Figure 6.5(a)), within which every two joints are connected, the instrument structure is relaxed into a tree graph (Figure 6.5(b)) with minimal number of connections. The tree graph can be further decomposed into a set of joint pairs, for which finding the optimal matching between two joint types is decided independently (Figure 6.5(c)), and the bipartite matching sub-problem then can be solved by maximum weight bipartite graph matching [240].

In our instrument structure configuration, there is N joints and M joint pairs, take the

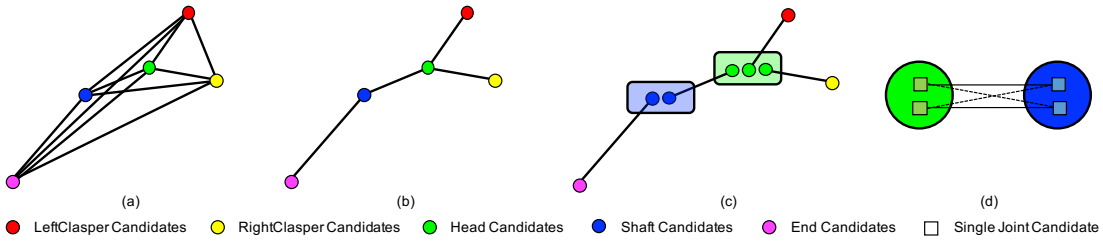


Figure 6.5: Graph relaxing for instrument structure: (a) Fully connected graph; (b) Tree structure graph; (c) A set of bipartite graphs after relaxation, the matching of joint pairs are decided independently. (d) Single joint pair connection example, multiple candidates are detected for each joint type, and the matching is solved by maximum weight bipartite graph matching.

m-th single joint pair C_m (e.g., the Head-Shaft joint pair and their association) as an example shown in Figure 6.5(d). Let $G_m = (D_{n_1} \bigcup D_{n_2}, E_m)$ be a bipartite graph, in which the joint candidates D_{n_1} and D_{n_2} are the nodes of the graph, and the edges E_m are all the possible pairs between the candidates of Head and Shaft joint. Besides, each edge is weighted by a given weight function based on the Head-Shaft part association map. The optimal matching of G_m is a subset of $M_m \subseteq E_m$ of the edges so that for each candidate $\mathbf{d}_n^k \in D_{n_1} \bigcup D_{n_2}$, there is at most one incident edge $e \in M_m$. The weight of the matching M_m is defined as the sum of the weights of the edges: $w(M) = \sum_{e \in M_m} w(e)$. So the minimum weight bipartite matching sub-problem in our application is to find an optimal matching of maximum weight for a given bipartite graph G_m and a given weight function. The optimal matching can be obtained using the classic Hungarian algorithm [241].

$$w(M_m) = \sum_{e \in M_m} w(e) = \sum_{\mathbf{d}^{k_1} \in D_{n_1}} \sum_{\mathbf{d}^{k_2} \in D_{n_2}} w_{k_1 k_2} \cdot c_{n_1 n_2}^{k_1 k_2} \tag{6.5}$$

$$\forall k_1 \in \{1 \dots K_{n1}\}, \sum_{\mathbf{d}^{k_2} \in D_{n_2}} c_{n_1 n_2}^{k_1 k_2} \leq 1 \tag{6.6}$$

$$\forall k_2 \in \{1 \dots K_{n2}\}, \sum_{\mathbf{d}^{k_1} \in D_{n_1}} c_{n_1 n_2}^{k_1 k_2} \leq 1 \tag{6.7}$$

where $w(M_m)$ is the overall weight of matching configuration for the m-th joint pair. To eliminate outliers and connect the right joints for each instrument, the association weight or score of joint candidate pairs $w_{k_1 k_2}$ is defined as the sum of accumulated pixel values along the line connecting the joint candidates $\mathbf{d}^{k_2}$ and $\mathbf{d}^{k_2}$ on the correspondent association heatmap. The association score of any possible joint candidate pair is used to construct the weighted bipartite graphs. Equation 6.6 and 6.7 set the constraint that no edges share the same node. Then, for all the joint pair, with the relaxation, the optimization is simply as the summation of individual joint pair:

$$\max w(M) = \sum_{m=1}^{M} \max w(M_m) \tag{6.8}$$

After finding the matching M_m with maximum score $w(M_m)$ of the chosen joint pairs, the ones which share the same joint can be assembled into full poses of multiple instruments.

6.6 Experiments and Results

6.6.1 Datasets and Analysis

Single-instrument Retinal Microsurgery Instrument Tracking (*RMIT*) Dataset This dataset[1] consists three image sequences during *in vivo* retinal microsurgery, with at most a single instrument in the field of view [148] and a resolution of 640×480 pixels. The statistics of the dataset is summarized in Table 6.3, and frame example from each sequence is shown in Figure 6.6. For each sequence, four joints (Tip1, Tip2, Shaft and End Joint) of the retinal instrument are annotated for most frames. Following the same training strategy as used in [150, 148, 149], the dataset is separated into training set including all the first halves of the sequences (577 frames), and test on the the second halves (578 frames).

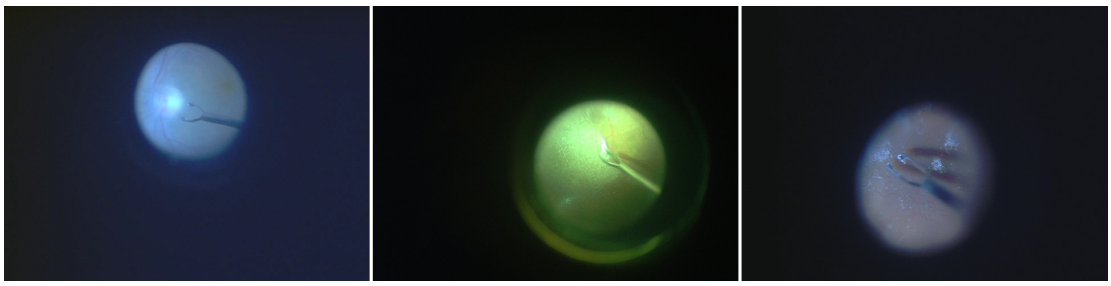


Figure 6.6: Example frame from each sequence of the single-instrument *RMIT* dataset.

Multi-instrument *EndoVis* Challenge Dataset For training and evaluating our network, we construct a high quality annotation using the *EndoVis* Challenge dataset[2]. The dataset is separated into training and test data: the training data includes four 45 seconds *ex vivo* video sequences of interventions (Seq 1-4), the test set is composed of 15 seconds additional video sequences for each of the training sequence (Seq 1-4), and two additional 1 minute recorded interventions (Seq 5-6). The frame resolution is 720×576 pixels.

For training and evaluating our network, we construct a high quality multi-joint annotation for this dataset. For each instrument, five joints including Left, Right Clasper, Head, Shaft and End joint are annotated. Compared to our multiple joint annotations, the original annotations only provide limited and non-intuitive pose information for training and testing purposes. We manually labelled 940 frames of the training data (4479 frames) and 910 frames for the test data (4495 frames). The statistics of the dataset is summarized in Table 6.3, and frame example from each sequence is shown in Figure 5.15. It is worth mentioning that in the additional video sequences in the test set there is a *EndoWrist* Curved Scissor instrument which does not appear in the training set.

The original and our proposed annotations are demonstrated in Figure 6.7 (a-b). The original annotation is retrieved from the robotic system, which includes the location of the intersection point between the instrument axis and the border between plastic and metal on the shaft, normalized Shaft-to-Head axis vector and the clasper angle. Compared to our multiple joint annotations, the original annotations only provide limited and non-intuitive pose information for training and testing purposes.

[1] https://sites.google.com/site/sznitr/code-and-datasets
[2] https://endovissub-instrument.grand-challenge.org/

	Seq - 1	Seq - 2	Seq - 3	Seq - 4	Seq - 5	Seq - 6	Whole
				***RMIT* Dataset**			
Train	201 / 201	111 / 111	265 / 271	-	-	-	577 / 583
Test	201 / 201	111 / 111	266 / 276	-	-	-	578 / 588
				***EndoVis* Dataset**			
Train	210 / 1107	240 / 1125	252 / 1124	238 / 1123	-	-	940 / 4479
Test	80 / 370	76/375	76 / 375	76 / 375	301 / 1500	301 / 1500	910 / 4495

Table 6.3: Label / Frame Number of the *EndoVis* and *RMIT* Dataset

To test the performance against noise, we also add Fractional Brownian Motion random noise on the test data in order to simulate smoke effect during surgery (see Figure 6.7 (c-d)).

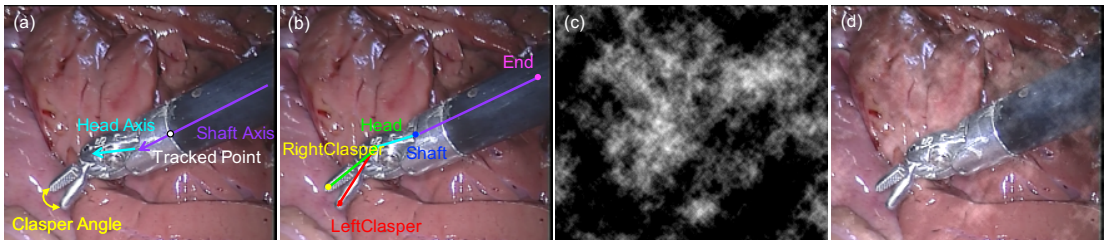


Figure 6.7: The (a) original and (b) original annotation for *EndoVis* dataset, (c) smoke effect simulation and (d) simulation overlaid on the frame.

Multi-instrument *In Vivo* Dataset Additionally, to test the framework performance on the *in vivo* data, we labelled 123 frames of video clips (1220 frames) which are obtained from robotic prostatectomy surgery conducted at University College London Hospitals NHS Foundation Trust (UCLH) with resolution of 1920×1080 pixels. Some of the frames are shown in Figure 6.8.

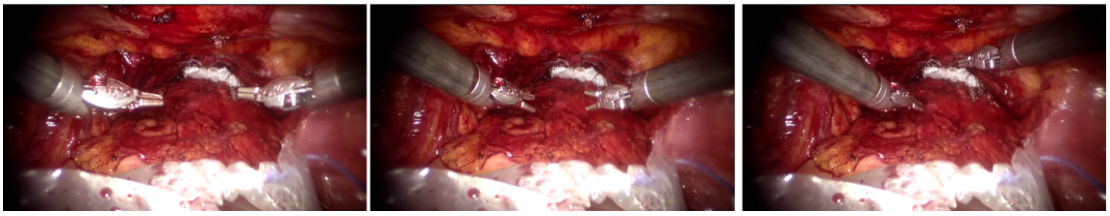


Figure 6.8: Example frames from the multi-instrument *in vivo* dataset.

Training and Runtime Analysis We implement our framework in Lua and Torch7[3]. The training data is augmented by horizontal and vertical flipping, and is resized to 256×320 px for *EndoVis* dataset and to 288×388 for *RMIT* dataset to fit in GPU memory. The detection radius r_d is set to 10 pixels for RMIT data, and to 15 pixels for *EndoVis* and *in vivo* data. The regression standard deviation σ is set to 20 pixels. The radius of NMS is set to equal the detection radius r_d. The network is trained on a single Nvidia GeForce GTX Titan X GPU using

[3] `http://torch.ch/`

stochastic gradient descent with an initial learning rate of 0.001 and momentum of 0.98. The learning rate progressively decreases every 10 epochs by 5%. The processing speed achieves 8.7 fps for videos, with the network inferencing taking 24 ms and the multi-instrument parsing step taking 89 ms.

6.6.2 *RMIT* Experiments

We trained the network with all four joints and we report performance by two different metrics: the Root-Mean-Square (RMS) distance (pixels) [148] and the strict Percentage of Correct Parts (strict PCP) [242]. The RMS distance reflects the localization accuracy of a single joint, it is evaluated as correct if the estimated joint location and the ground truth is within the threshold. Meanwhile the strict PCP estimates the localization of a joint pair and is considered correct if the distances between two connected joints are both smaller than α times the ground truth length of the connection pair. The evaluation results are shown in Table 6.4 and Table 6.5. We report the average RMS error distance, only on frames which the instruments are correctly detected (within the threshold measure). The same criteria apply for other datasets evaluated in the paper. We also compared the result against the state-of-the-art methods in Table 6.6 and Table 6.7. In previous papers as listed in Table 6.6 and Table 6.7, only recall score is reported. Approximate numbers are obtained through the accuracy threshold graphs from the papers, which do not provide the precise number. Analogously to previous methods, the recall score is evaluated by means of threshold measure (15 pixels) for the separate joint of the pose predictions and α for strict PCP is set to 0.5.

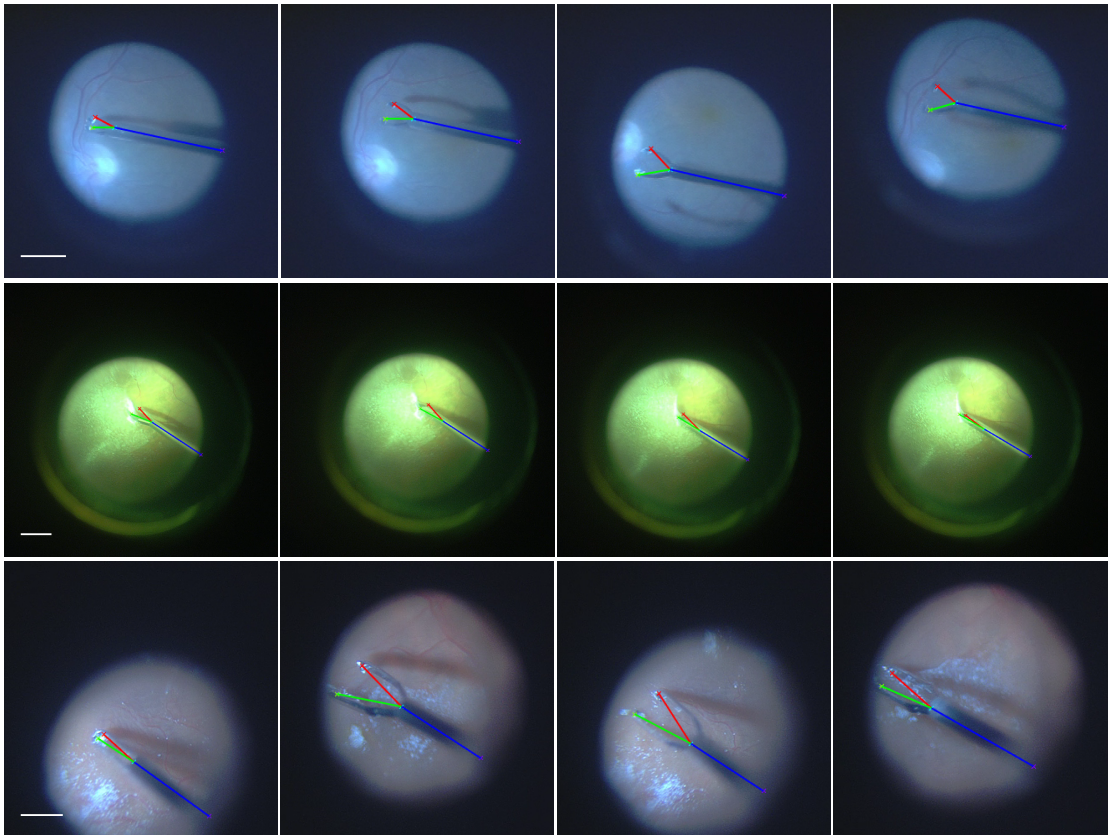


Figure 6.9: Examples of the single-instrument *RMIT* dataset. The frames are trimmed around the instrument for better visualization. Scale bar equals 50 pixels.

Recall (%) / Precision (%) / Distance (px) of the *RMIT* Dataset ($Thres = 15$ px)					
	Tip1	**Tip2**	**Shaft**	**End**	**Total**
			Train set		
Reca.(%)	100.0	100.0	100.0	100.0	100.0
Prec.(%)	100.0	100.0	100.0	100.0	100.0
Dist.(px)	2.14	2.28	1.72	2.38	2.13
			Test set		
Reca.(%)	99.13	97.58	94.12	86.51	94.33
Prec.(%)	99.13	97.58	94.12	86.51	94.33
Dist.(px)	5.26	4.61	4.93	4.68	4.87

Table 6.4: Quantitative Results of the RMIT Dataset: Precision and the Distance Error Between Ground Truth and the Estimate of Each Joint. The Threshold is Set to 15 Pixels for the Original Resolution of 640×480 Pixels.

Recall (%) / Precision (%) for Strict PCP of the *RMIT* Dataset ($\alpha = 0.5$)				
	Tip1-Shaft	**Tip2-Shaft**	**Shaft-End**	**Total**
		Train set		
Reca.(%)	100.0	100.0	100.0	100.0
Prec.(%)	100.0	100.0	100.0	100.0
		Test set		
Reca.(%)	99.13	97.58	94.12	96.94
Prec.(%)	99.13	97.58	94.12	96.94

Table 6.5: Quantitative Results of the RMIT Dataset: the Strict PCP Score of the Estimate of Each Joint Pair.

For the proposed methods, the average joint distance error for the test set is 4.87 pixels with the same recall and precision score of 94.33%, and the average strict PCP recall score is 96.94%. Some of the test set results are shown in Figure 6.9. Even under different lighting conditions, the model can predict the pose of the instrument correctly. It is interesting to point out that even though the association map used is constructed using a straight line, it still works on titled instruments (see the bottom line of Figure 6.9 for example). This implies that the rectangle association maps are learnt to indicate the connection relationships between joint pairs. The trained network predicts joint pair connections by not only relying on the instrument pixels, but also on the learnt joint relations and spatial contextual information.

As we listed in Table 6.6, previous methods mainly focus on the evaluation of Shaft joint, except for SRNet [243], where our performance is on par with SRNet. The recall score of the End joint is the lowest (86.51%) among the four joints, due to its ambiguous annotation and image blur. SRNet uses a different strategy by explicitly modelling the instrument joints and their presence, which simultaneously predicts the instrument number and their pose. By

[3]To maintain notation consistency, the Shaft and End joint in our paper correspond respectively to End Shaft and Start Shaft joint in previous papers.

[4]To maintain notation consistency, the Shaft and End joint in our paper correspond respectively to End Shaft and Start Shaft joint in previous papers.

The Recall Score of the *RMIT* Test Set ($Thres = 15$ px)					
	Tip1	**Tip2**	**Shaft**	**End**	**Total**
DDVT [148]	-	-	< 85.0	-	-
POSE [149]	-	-	< 90.0	-	-
RTOA [150]	-	-	≈ 90.0	-	-
SRNet [243]	98.6	94.1	96.2	91.2	95.0
Proposed	99.1	97.6	94.1	86.5	94.3

Table 6.6: Quantitative Recall Performance Comparison with the State-of-the-art Methods on the RMIT Test Set[4]

The Strict PCP Score of the *RMIT* Test Set ($\alpha = 0.5$)				
	Tip1-Shaft	**Tip2-Shaft**	**Shaft-End**	**Total**
POSE [149]	≈ 95.0	≈ 90.0	-	-
Proposed	99.13	97.58	94.12	96.94

Table 6.7: Quantitative Strict PCP Score Comparison with the State-of-the-art Methods on the RMIT Test Set

assuming a known maximum number of instrument in the field of view, it bypasses the joint detection and association two-stage process, so can be trained in an end-to-end fashion. Adding prior could help constrain the problem, compared to SRNet, we want to treat the task as general as possible, so our model does not rely on any prior knowledge of the number of instrument, theoretically it can predict pose of arbitrary number of instrument, which one of the potential strengths of our framework.

6.6.3 *EndoVis* Experiments

Since our annotation is limited, we used our network with five joints using all the training data generated from high quality our annotation. We reported the average precision, recall score and RMS distance (pixels) of each joint for all the test data in Table 6.8. With a threshold of 20 pixels for the original resolution of 720×576 pixels, the average joint distance error for the test data set is 6.96 pixels with a recall score of 82.99% and a precision score of 83.70%.

Similar to the *RMIT* dataset result, the lower score for the End joint ($76.81\%/77.71\%$) is reasonable since it does not have distinct features and even the manual annotation has high variance. If the threshold is relaxed to 30 pixels, the recall and precision score of the End joint increase to 89.78% and 90.68% respectively. For the Head joint with the lowest recall and precision ($75.82\%/76.81\%$) in Table 6.8 we separate the test dataset results into Test set of sequence 1-4, which is seen in the training data, and Test set of sequence 5-6, which are the two additional sequences. As we have mentioned before, the two additional sequences exhibit a Curved Scissor instrument which is not seen in the training set. In Figure 6.10 and Figure 6.11, we show some pose estimation examples from the test set. As we can see, the left *EndoWrist* Curved Scissor instrument has a different shape compared to the right *EndoWrist* Needle Driver instrument, which explains the relatively low score especially for the Head joint. But our model is general enough to detect individual parts of this new instrument. Clearly,

EndoVis Dataset						
	LeftClasper	**RightClasper**	**Head**	**Shaft**	**End**	**Total**
			Train set ($Thres = 20$ px)			
Reca. (%)	100.0	100.0	99.57	100.0	99.89	99.89
Prec. (%)	99.95	99.95	99.68	99.95	99.84	99.87
Dist. (px)	2.43	2.53	2.34	2.74	6.73	3.36
			Test set Seq 1-4 ($Thres = 20$ px)			
Reca. (%)	94.64	86.20	97.56	100.0	89.77	93.64
Prec. (%)	94.64	86.20	97.56	100.0	89.77	93.64
Dist. (px)	3.90	4.48	6.18	6.74	8.85	6.03
			Test set Seq 5-6 ($Thres = 20$ px)			
Reca. (%)	82.00	85.13	64.70	85.71	70.18	77.55
Prec. (%)	82.56	85.63	66.20	87.15	71.54	78.62
Dist. (px)	5.62	5.87	6.74	9.60	9.33	7.43
			Test set ($Thres = 20$ px)			
Reca. (%)	86.28	85.49	75.82	90.55	76.81	82.99
Prec. (%)	86.65	85.82	76.81	91.50	77.71	83.70
Dist. (px)	5.03	5.40	6.55	8.63	9.17	6.96
			Smoke Test set ($Thres = 20$ px)			
Reca. (%)	83.85	82.69	74.89	89.73	82.25	82.68
Prec. (%)	83.48	82.27	75.07	89.71	82.55	82.62
Dist. (px)	5.25	5.72	6.50	8.62	8.86	6.99
			Test set Seq 1-4 ($Thres = 30$ px)			
Reca. (%)	95.45	89.29	97.56	100.0	100.0	96.46
Prec. (%)	95.45	89.29	97.56	100.0	100.0	96.46
Dist. (px)	4.04	5.23	6.18	6.74	11.01	6.64
			Test set Seq 5-6 ($Thres = 30$ px)			
Reca. (%)	86.13	87.13	71.10	91.69	84.55	84.12
Prec. (%)	86.71	87.62	72.51	93.08	85.91	85.17
Dist. (px)	6.35	6.38	7.98	10.74	11.87	8.66
			Test set ($Thres = 30$ px)			
Reca. (%)	89.29	87.86	80.05	94.51	89.78	88.30
Prec. (%)	89.67	88.19	80.99	95.42	90.68	88.99
Dist. (px)	5.57	5.99	7.37	9.38	11.58	7.98
			Smoke Test set ($Thres = 30$ px)			
Reca. (%)	88.30	86.81	78.02	95.66	91.32	88.02
Prec. (%)	88.02	86.41	78.30	95.66	91.48	87.97
Dist. (px)	6.13	6.68	7.19	9.76	10.60	8.07

Table 6.8: Quantitative results of the *EndoVis* dataset: precision and the distance error between GT and the estimate of each joint. For the *EndoVis* dataset, the thresholds are set to 20 and 30 px for the original and smoke-simulated test data with the resolution of 720×576 px.

the generalisation to an unseen new instrument is limited to certain degree. Although the left Curved Scissor instrument has different appearance, it shares the same joint configuration with the Needle Driver instrument. We observe that our model works well on self occlusion, as

shown in the first row of Figure 6.11. This is credited to: (I) the model learns the spatial relationship between joints, even if a joint is occluded, it can be inferred from other joints; (II) the training data contains self occlusion examples that can be used by the model for handling self occlusion. The results we display show that with limited training data, our model is still capable of generalising to some degree. From Figure 6.12 and Table 6.8 we can also see that under smoke simulations the performance on test data only decrease slightly to 82.68% for recall and 82.62% for precision, with distance errors of 6.99 pixels.

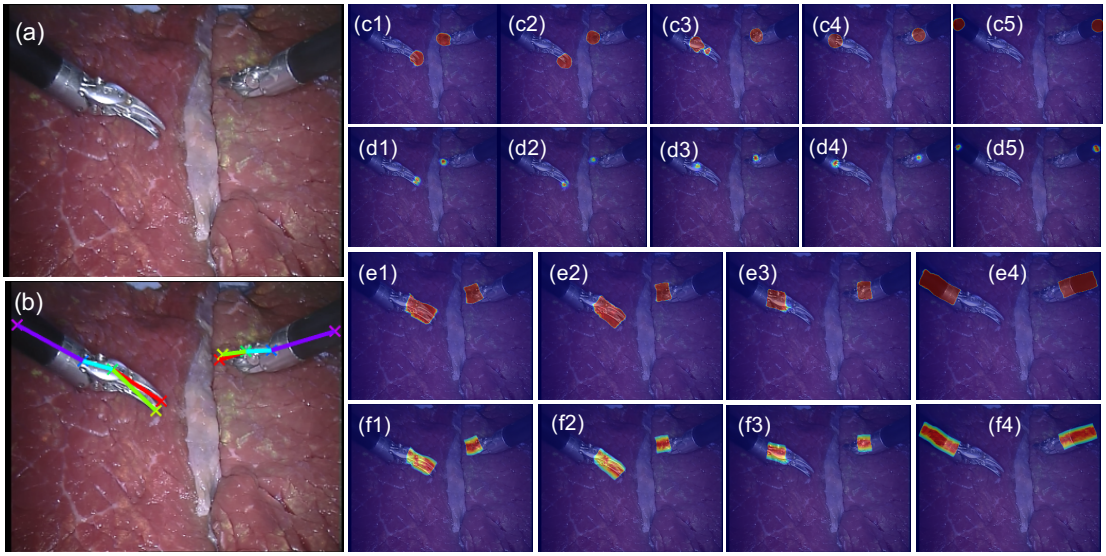


Figure 6.10: Result example from test set. (a) The original frame; (b) the estimated pose; joint (c1-5) and association (d1-5) score output from detection subnetwork; joint (e1-4) and association (f1-4) output from regression subnetwork.

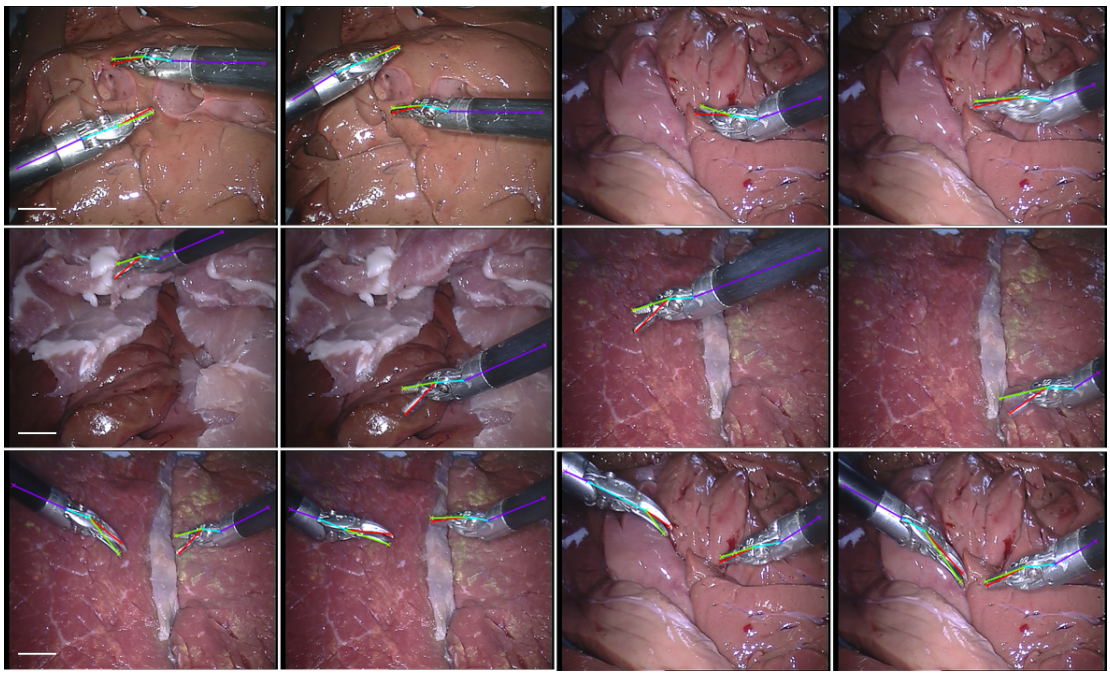


Figure 6.11: Examples of original *EndoVis* test data. Our network is able to detect a new instrument (Curved Scissor) which are not seen in the training data. Scale bar equals 100 pixels.

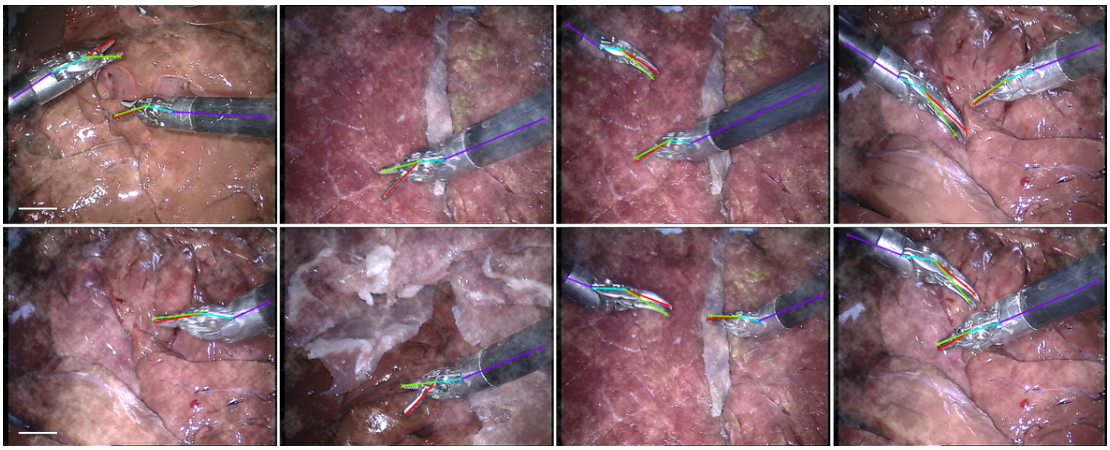


Figure 6.12: Examples of smoke-simulated *EndoVis* test data. Our network is able to detects instruments which are not seen in the training data even under smoke simulation. Scale bar equals 100 pixels.

Finally, we use *EndoVis* dataset to perform an ablation study[5] to understand the detection-regression architecture. We compared the performance of five different models, including detection-only, shallow regression-only, deep regression-only, single-branch detection-regression and our proposed bi-branch detection-regression model. For the detection-only model, we use the output probability maps from the detection subnetwork for direct pose estimation. We also trained two regression-only models, a shallow one with the same architecture as the regression submodule in our detection-regression model and the input is the RGB frame without the detection probability maps, the deep one whose architecture is the same as the detection-only model and with Gaussian regression ground truth. For the single-branch model, we fuse two branches of the detection submodule into only one branch with double size of the feature maps of our model. The performance comparison of different models is summarized in Table 6.9. The bad performance of the detection-only model ($32.19\%/14.41\%$ for recall and precision score) is expected. As seen from the ground truth binary map in Figure 6.3, the pixels belonging to the joint have the same weight, which lead to bad localization of joints. We also observe that both regression-only models have better performance. It is interesting that the precision score for deep model (97.67%) is higher than that for the shallow model (71.53%), while either shallow or deep regression-only models achieve similar recall performance (66.46% for shallow model and 65.06% for the deeper model). Deeper architecture does not help to achieve better recall performance in the experiment. We infer that one of the reasons is that the size of the training data is relatively small, which affects model generalization. The regression-only models are capable of predicting the location of joints without any guidance. However, regression is empirically too localized, which supports small spatial context [97], the process of regressing from original input image to joint location directly can be difficult. By combining detection and regression, the detection module guides where to focus and provides spatial contextual information between joints for the regression module, by using the probability output from the detection module as structural guidance, the regression module facilitates the detection module to localize the joints more precisely. The performance of both

[5] An ablation study refers to evaluating how the performance is affected by removing some part of the model.

***EndoVis* Test Set** *(Thres = 20 px)*						
	LeftClasper	**RightClasper**	**Head**	**Shaft**	**End**	**Total**
Detection-only Network						
Reca. (%)	32.58	24.51	29.40	40.27	34.18	32.19
Prec. (%)	14.28	11.05	13.19	18.03	15.49	14.41
Dist. (px)	7.94	6.22	6.75	8.87	7.57	7.47
Shallow Regression-only Network						
Reca. (%)	67.73	81.26	66.48	75.16	41.65	66.46
Prec. (%)	72.94	84.49	73.35	80.65	46.23	71.53
Dist. (px)	4.86	4.34	6.18	7.58	9.06	6.41
Deep Regression-only Network						
Reca. (%)	65.75	61.81	66.48	66.65	64.62	65.06
Prec. (%)	98.79	93.35	99.34	99.40	97.47	97.67
Dist. (px)	3.63	3.80	5.12	6.84	7.15	5.31
Single-branch Detection-Regression Network						
Reca. (%)	78.90	81.04	74.07	79.56	70.27	76.77
Prec. (%)	88.13	90.27	83.74	88.94	79.71	86.16
Dist. (px)	4.70	5.44	7.24	7.72	9.22	6.87
Proposed Detection-Regression Network						
Reca. (%)	86.28	85.49	75.82	90.55	76.81	82.99
Prec. (%)	86.65	85.82	76.81	91.50	77.71	83.70
Dist. (px)	5.03	5.40	6.55	8.63	9.17	6.96

Table 6.9: Ablation Study of the Detection-Regression Model Architecture on EndoVis Test Set

detection-regression models shows the improvement, and furthermore, our network takes less time to train compared to regression-only model. The single-branch model achieves the performance of $76.77\%/86.16\%$ for recall and precision, which is nearly as good as the bi-branch model. We would like to point out that single-branch and bi-branch models are essentially similar. We choose bi-branch architecture here to conceptually separate the training of joint and joint association into two branches.

In Figure 6.13, we have presented two failure cases on the test set. When one instrument is occluded by another one (Figure 6.13 (a)), the model cannot infer the occluded joints, we think it is due to the lack of training data on instrument overlap, which causes the model fail to learn or handle the complex situation. We can compare this to the self-occlusion (first row of Figure 6.11). Since the training data covers self-occlusion, the model can well detect the self-occluded joints. We also show in Figure 6.13 (b) that some joints of the new Curved Scissor instrument are not well localized, e.g. the Head joint. Our model has extended certain generalizability to unseen instrument, but obviously compared to the Needle Driver instrument in the training data, the performance is less robust.

6.6.4 *In Vivo* Experiments

We fine-tuned the above trained model on 80% of the labelled data (97 frames) with a fixed learning rate 0.0001 for 10 epochs and tested on the whole sequences. In Table. 6.10, The *in vivo* video sequence we use is with high resolution 1920×1080 pixels, so we set the threshold

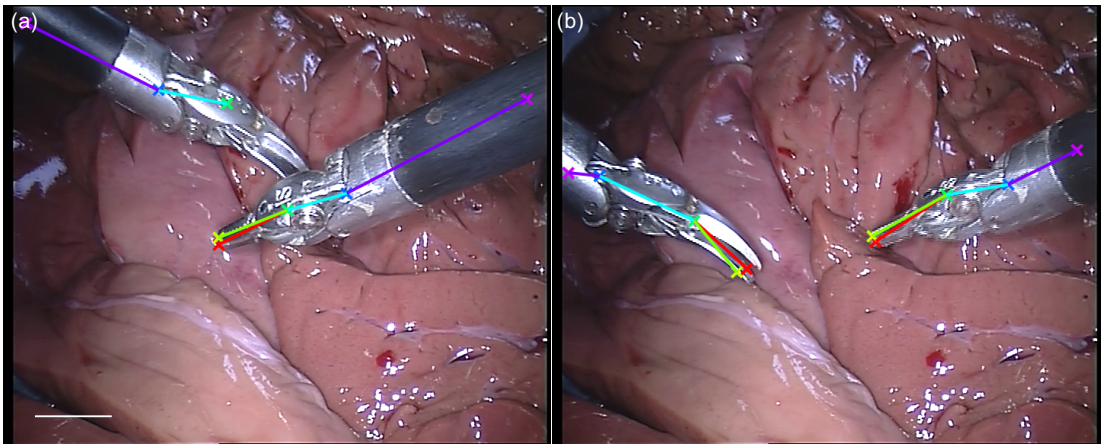


Figure 6.13: Examples of failure cases of *EndoVis* test set. (a) Occuded joints are miss detected; (b) The head joint of the new Curved Scissor instrument on the left is not well localized. Scale bar equals 100 pixels.

as 50 pixels for evaluation. In Table 6.10, it is shown that the average distance errors are reduced to 9.81 and 13.42 pixels for the train and validation set respectively, with the threshold of 50 pixels for the original resolution. Examples of the *in vivo* data are shown in Figure 6.14. We did not perform any temporal processing operations in order to present the authentic results.

***In Vivo* Dataset** ($Thres = 50$ px)						
	LeftClasper	**RightClasper**	**Head**	**Shaft**	**End**	**Total**
			Train set			
Reca.(%)	97.94	97.94	100.0	100.0	98.97	98.97
Prec.(%)	96.39	96.39	98.97	100.0	98.97	98.14
Dist.(px)	7.84	8.40	9.61	10.39	12.81	9.81
			Validation set			
Reca.(%)	98.08	94.23	96.15	100.0	92.31	96.15
Prec.(%)	96.15	92.31	94.23	100.0	92.31	95.00
Dist.(px)	13.91	12.54	12.01	13.86	14.77	13.42

Table 6.10: Quantitative results of the *in vivo* dataset: precision and the distance error between GT and the estimate of each joint. For the *in vivo* data, the threshold is set to 50 px for the original resolution of 1920×1080 px.

6.7 Discussion

In this chapter, we have proposed a deep neural network based 2D pose estimation framework for articulated multiple articulated instruments in surgical images and video. The methodology performs detection of the instruments and their degrees of freedom without using kinematic information from robotic encoders or external tracking sensors. To the best of our knowledge, it represents a novel attempt to perform image-based articulated pose estimation at this level of detail and can potentially be extended to handle even more complicated flexible articulation by incorporating additional joint nodes. In our approach, joints and associations between joint pairs are first detected and then refined in a detection-regression FCN. To obtain the final pose

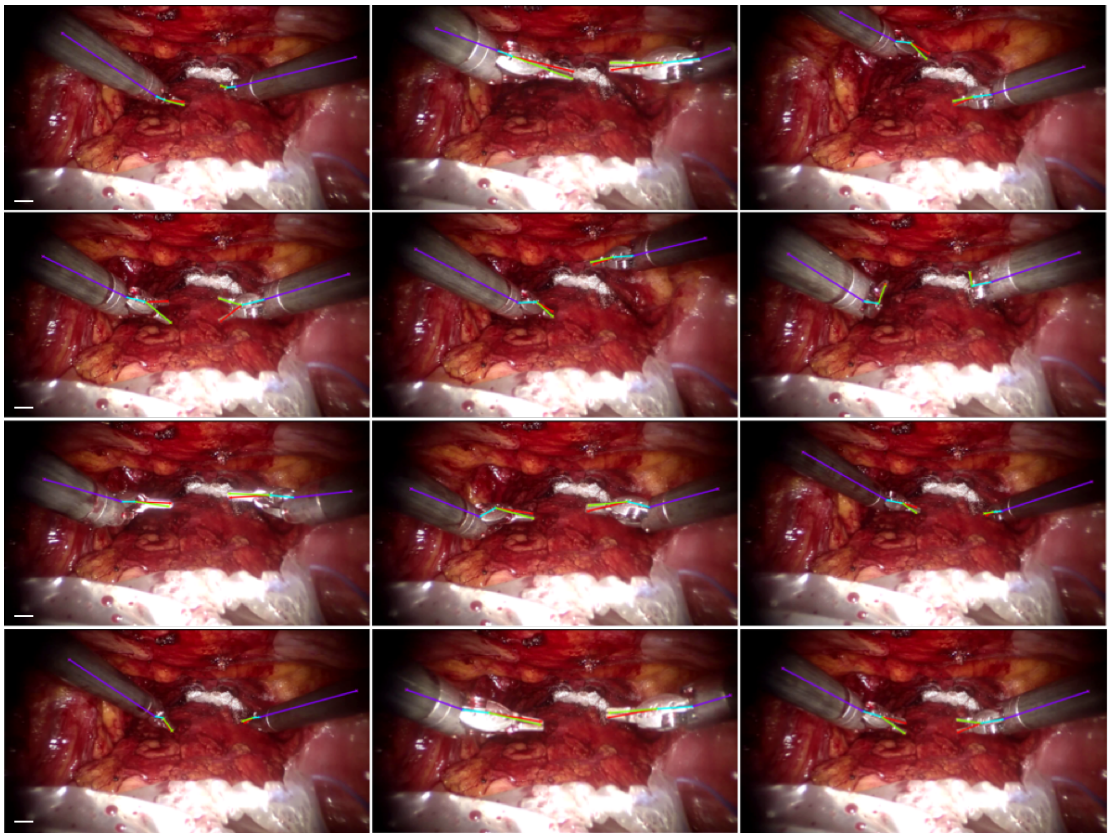


Figure 6.14: Examples of *in vivo* data with our fine-tuned model. The results demonstrate the capacity of our framework to be applied to real surgical scenes. Scale bar equals 100 pixels.

of all the instruments in an image, association heatmaps are used as a measurement to connect joint pairs for each instrument by maximum bipartite matching. The framework has been trained and evaluated on single-instrument *RMIT* dataset, and multi-instrument *ex vivo EndoVis* and *in vivo* datasets with detailed annotations adding to existing challenge data. Interestingly, our experiments show that our model exhibits some generalizability to new unseen instrument and has good robustness under smoke simulation. The performance on the *in vivo* datasets demonstrates the capacity of our framework to handle real surgical scenes. The performance on the *in vivo* datasets demonstrates the capacity of our framework to handle real surgical scenes. The dataset annotations and our model will be publicly released with our model to support research in the field[6]. A current limitation of our method is that it is limited to 2D inference and a natural extension would be to explore the estimation of 3D articulation. This seems plausible when using stereo configurations which are available within the *EndoVis* data for example and can potentially be used to formulate both the detection and the pose estimation in a joint space of both views. Additionally, it will be interesting to explore the sequential tracking of articulated instruments. This could potentially be achieved by probing the motion information that can be learnt through recurrent neural networks.

[6] `https://github.com/surgical-vision/EndoVisPoseAnnotation`

Chapter 7

Conclusion and Perspectives on Future Research Possibilities

7.1 Overview of Thesis and Scientific Contributions

This thesis has presented methods for visual tracking in MIS using images from endoscopic and laparoscopic cameras. The methods provide solutions to problems involving tissue surface deformation estimation, surgical instrument tracking and instrument pose estimation. The developed vision-based methods can be incorporated with computer assisted interventions where the surgeon is supported by advanced computing systems which simplify surgical tasks or enable new surgical capabilities.

With the increasing use of MIS for many surgical procedures, the developed algorithms can contribute to solving current challenges during surgery. Tissue motion can complicate surgical dexterity and impede intra-operative imaging, therefore, tissue surface deformation estimation can be used to apply motion stabilization using robotics or prescribe dynamic constraints to avoid critical anatomical structures through a robotic-assisted system. Similarly, effective intra-operative instrument localisation or pose estimation can provide safer surgery by avoiding vulnerable structures or generate precise measurements of the surgeon's moving patterns for surgical skill training or post-surgery analysis.

In Chapter 3, we propose a hybrid non-rigid tissue surface deformation estimation method which represents the tracked surface with a geometric mesh model, combining sparse feature tracking and dense intensity-based method. Our hybrid method improves the capability of tracking because our intensity-based component replaces the original SSD metric with the SCV metric to facilitate tracking under dramatic illumination variations. Because intensity based methods are prone to failure without a good initialization, our feature tracking component guides the optimization towards correct convergence when tissue dynamics undergoes large intra-frame displacements, allowing the following pixel-wise dense tracking to refine the result. We show that the proposed method can be used in practice to track tissue and allow better multispectral imaging that compensates for tissue dynamics.

In Chapter 4, we demonstrate a keypoint based 2D tracker which facilitates a pre-existing 3D tracker for robust surgical instrument tracking. The 3D tracker usually gradually drifts away when the instrument motion is fast or complex, or when the instrument is out-of-view, since the assumption is that the parameter search is restricted to a local region from the last frame. Our 2D

keypoint-based tracker relies on a novel rotation-invariant GHT, enabling a global search of the entire frame for the tracked instrument, the 2D tracking adapts with a histogram probabilistic segmentation model, which corrects the drift by providing the 3D tracker a good initialization for each frame. The combination of our 2D and 3D tracking is proved to be helpful for long term robust surgical instrument tracking.

In Chapter 5, we introduce our PAWSS tracker, which tackles the tracking problem from the tracking-by-detection point of view and can be used to track both tissue and instrument targets. A tracking-by-detection tracking framework considers the tracking task as a binary classification problem. Given the object location initialized with a bounding box in the first frame, the tracked object appearance model is represented by patch-wise descriptors. To suppress background information in the bounding box, each patch is weighted using the probabilistic segmentation model, with object patches associated higher weights and background patches associated lower weights. Then, the classifier detects and updates the appearance model online using structured output SVM learning framework with positive and negative samples in the consecutive frames. The tracking framework generalize to any 2D tracking task and works well for *in vivo* surgical instrument tracking.

In Chapter 6, we exploit the deep learning paradigm for multiple articulated instrument pose estimation. The limited availability of annotated datasets is one of the obstacles for supervised deep learning techniques applied in MIS video and images. For training our network, we construct high quality annotations for an existing endoscopic challenge dataset with multiple joint labelling. Articulated pose is represented by the position of each individual joint part of the instrument. For multiple instruments, joints belonging to the same instrument also need to be correctly associated. We propose the detection-regression FCN, which only relies on vision cues rather than CAD models or robotic kinematic information. Instead of employing low-level complicated feature engineering, the raw pixels of the entire image are fed into the deep multi-layer network, after training, heatmaps with different detected joints and with association between joint pairs are obtained, the output are used as measurements for maximum bipartite matching to infer the full poses of multiple instruments. Our annotations and model provide an excellent baseline for future research in this field and for comparative studies.

7.2 Challenges and Limitations

While the proposed solutions for different visual tracking in endoscopic MIS are promising, there are a number of limitations that restrict applicability and robustness or the intrinsic capability of our algorithms.

7.2.1 Tissue Tracking

One of the limitations of our non-rigid surface deformation estimation method is that it cannot be applied to the entire field of view of the surgical site, which may be composed of different anatomical structures or multiple organs. As shown in Figure 3.1, with our method the tracked surface is represented by a triangular geometric mesh model in a sub-section of the image space and it would require extension to be applied to multiple parts of the image. Additionally, the 2D planar mesh model cannot reflect the real depth of the surface as it only tracks in 2D without considering the 3D real nature of the surface. Also, during optimization the deformation of the

mesh is constrained by the regularization term to prevent the mesh topology from wrinkling too excessively. The stronger the regularization term is, the more rigid the deformation estimate is and vice versa. The regularization factor is usually set manually based on empirical evidence, which means the model does not adapt to various organ-specific tissue elasticity and motion pattern. Also, the mesh model is generalized to piecewise affine warp transformation with the coordinates of mesh vertices being employed as the parameters of the algorithm. Therefore, to represent complex tissue motion, increasing the number of mesh vertices can increase the freedom, but also increases the algorithm complexity. Besides, the approach does not naturally cope with fully occlusion or out-of-view situation.

7.2.2 Robust Instrument Tracking

GHT extends the well-known Hough Transform to detect arbitrary shapes by describing shapes as collections of spatial features in a local coordinate system. Given an example image containing the object of interest, the feature orientation and the relative displacement and orientation to the reference point are computed and stored to fully represent the target object in our GHT-based tracker. The sparse keypoint feature representation enables real-time performance but the limitation is that the detection of features is not always robust. Additionally, there is a requirement to know the geometry of the object in advance and the feature positions on it. They are prone to fail for texture-less target for not having enough or well-distributed features. Besides, the voting strategy used to locate the target is designed to handle any abrupt in-plane rotation, which we refer to as rotation-invariant hough voting scheme. However, it is potentially not resilient at handling abrupt out-of-plane rotation. When the tracked features are rotated dramatically out of view, the distribution of the features will be different from the target template. So, the algorithm handles out-of-plane rotation the same way as appearance changes by template update or evoking the detection module.

The tracking-by-detection method we developed (PAWSS) has some favourable characteristics that avoid explicit keypoint detection. Results compared to the generative method are more appealing, especially for long term tracking where appearance variation may occur. However, the PAWSS classifier style approach usually employs local search mechanisms, which assumes the target is near the previous location. A possible solution to extend this is to add a target detection module in the framework by feature-based or sliding window-based approaches. Sparse keypoint features are detected and used as the target representation alike to our GHT method. The target template is then matched in the following frames by global search. For the sliding window-based methods, the input frame is scanned by a window of various sizes, and each window patch is classified whether containing the target or not. For example, TLD [3] employed a novel tracking-by-detection framework, but it performs well in long sequences by introducing a sliding window-based detector with cascaded architecture.

7.2.3 Instrument Pose Estimation

Supervised deep learning based methods rely on the availability of large scale training data. For our multi-instrument 2D pose estimation framework, one of the limitations is that in vision for MIS, there are not many large scale annotated endoscopic video datasets available, and the *EndoVis* dataset we use is a small dataset, which only includes one specific instrument, the da Vinci

EndoWrist long needle driver, and does not include enough instrument occlusion examples. As this is the only public dataset available for endoscopic instruments, our trained network works well at recognising such instruments, but generally speaking, it has limited capacity in detecting various kinds of endoscopic instruments and is prone to over-fitting. It is then predictable that it is difficult for the trained model to detect instruments correctly under a wide range of variations. This limitation can be overcome given enough data. Besides, as the framework is not an end-to-end framework, the network output is the detection of individual parts of the instrument, following the non-maximum suppression step to generate joint candidates and the multi-instrument parsing step to associate correct joints belongs to the same instrument. The association score is accumulated along the possible joint candidate pair, when inferring the full poses, thresholds are set manually to exclude invalid joint candidate or joint pair association. For certain frames, it is possible that certain joints (e.g., occluded joints) in the output heatmap provide low confidence. Since the thresholds are fixed for any input frame, joints may be missed or erroneously localized, which affect the final estimation. The ideal solution will be training an instance-level network which can automatically recognize the number of instruments in the frame and the joints from each instrument. Such training data is likely to become more widely available with more precise annotation efforts already in progress.

7.3 Future Research Directions

The body of work presented in this thesis can naturally be extended. For surgical instrument localization and especially for pose estimation, one of the technical challenges is instrument occlusion. During surgery, instruments can be partially occluded by tissue or other instruments or even completely leave the field of view. Tracking-by-detection methods usually suffer from drift when the tracked object is occluded or disappears for a long period of time. This is related to the model update component in the tracking system because background information will be included in the object appearance, which will eventually lead to tracking drift or failure. The problem can be tackled by adding extra re-detection components, for example by maintaining the original object template, so the object will be re-detected when it is lost, albeit such solutions are not always effective. Besides, occluded parts of the instrument can be predicted by the visual parts based on its shape or skeleton. The structural constraints provide indication to filter out implausible poses and achieve better prediction outputs.

Our current pose estimation framework of articulated instrument is single-frame based, which does not explore any motion information. While these pure visual based methods have gained great success, they highly relied on the availability of large scale training data, and one of the bottlenecks is the generation of manual labelling, that can be time consuming and labour intensive. Therefore, an interesting possible research direction is to explore articulated motion patterns of the instruments from the kinematic information. The hope is that motion patterns can act as a strong and complementary supervision to pose appearance, and the ground truth can be easily obtained in an automated fashion. Combining temporal motion clues, topological or kinematic constraints with visual features provides more valuable insights into understanding of the surgical environment, which will be beneficial to high level tasks such as surgical workflow recognition, automated skill assessment and robotic instrument manipulation, etc.

Finally, an exciting possible development is to fuse methodologies for instrument and tissue tracking and providing a comprehensive joint framework for tracking all motion within the surgical site. Combining our surface tracking approach and an instrument tracker is one direct extension possibility. Implementing this in 3D would provide a detailed scene flow for the entire image space and incorporating additional sensors information, for example robotic instrument kinematic data, would inject robustness into the motion field maps.

Bibliography

[1] G. Nebehay and R. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pp. 862–869, IEEE, 2014.

[2] H.-U. Kim, D.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "Sowp: Spatially ordered and weighted patch descriptor for visual tracking," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 3011–3019, IEEE, 2015.

[3] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.

[4] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1838–1845, IEEE, 2012.

[5] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 263–270, IEEE, 2011.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision (ECCV), 2012 European Conference on*, pp. 702–715, Springer, 2012.

[7] H. Hopkins and N. Kapany, "A flexible fibrescope, using static scanning," *Nature*, vol. 173, no. 4392, pp. 39–41, 1954.

[8] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic surgery: a current perspective," *Annals of Surgery*, vol. 239, no. 1, p. 14, 2004.

[9] J. H. Kaouk, W. M. White, R. K. Goel, S. Brethauer, S. Crouzet, R. R. Rackley, C. Moore, M. S. Ingber, and G.-P. Haber, "Notes transvaginal nephrectomy: first human experience," *Urology*, vol. 74, no. 1, pp. 5–8, 2009.

[10] R. Autorino, R. J. Stein, E. Lima, R. Damiano, R. Khanna, G.-P. Haber, M. A. White, and J. H. Kaouk, "Current status and future perspectives in laparoendoscopic single-site and natural orifice transluminal endoscopic urological surgery," *International Journal of Urology*, vol. 17, no. 5, pp. 410–431, 2010.

[11] V. Vitiello, S.-L. Lee, T. P. Cundy, and G.-Z. Yang, "Emerging robotic platforms for minimally invasive surgery," *IEEE Reviews in Biomedical Engineering*, vol. 6, pp. 111–126, 2013.

[12] C. Tsui, R. Klein, and M. Garabrant, "Minimally invasive surgery: national trends in adoption and future directions for hospital strategy," *Surgical Endoscopy*, vol. 27, no. 7, pp. 2253–2257, 2013.

[13] D. J. Mirota, M. Ishii, and G. D. Hager, "Vision-based navigation in image-guided interventions," *Annual Review of Biomedical Engineering*, vol. 13, pp. 297–319, 2011.

[14] L. Maier-Hein, S. S. Vedula, S. Speidel, N. Navab, R. Kikinis, A. Park, M. Eisenmann, H. Feussner, G. Forestier, S. Giannarou, *et al.*, "Surgical data science for next-generation interventions," *Nature Biomedical Engineering*, vol. 1, no. 9, p. 691, 2017.

[15] S. Vidal-Sicart, R. V. Olmos, O. Nieweg, R. Faccini, M. Grootendorst, H. Wester, N. Navab, B. Vojnovic, H. van der Poel, S. Martínez-Román, *et al.*, "From interventionist imaging to intraoperative guidance: New perspectives by combining advanced tools and navigation with radio-guided surgery," *Revista Española de Medicina Nuclear e Imagen Molecular (English Edition)*, vol. 37, no. 1, pp. 28–40, 2018.

[16] M. E. Ivan, J. Yarlagadda, A. P. Saxena, A. J. Martin, P. A. Starr, W. K. Sootsman, and P. S. Larson, "Brain shift during bur hole–based procedures using interventional mri: Clinical article," *Journal of Neurosurgery*, vol. 121, no. 1, pp. 149–160, 2014.

[17] M. J. White, J. S. Thornton, D. J. Hawkes, D. L. Hill, N. Kitchen, L. Mancini, A. W. McEvoy, R. Razavi, S. Wilson, T. Yousry, *et al.*, "Design, operation, and safety of single-room interventional mri suites: Practical experience from two centers," *Journal of Magnetic Resonance Imaging*, vol. 41, no. 1, pp. 34–43, 2015.

[18] E. R. McVeigh, M. A. Guttman, R. J. Lederman, M. Li, O. Kocaturk, T. Hunt, S. Kozlov, and K. A. Horvath, "Real-time interactive mri-guided cardiac surgery: Aortic valve replacement using a direct apical approach," *Magnetic Resonance in Medicine*, vol. 56, no. 5, pp. 958–964, 2006.

[19] R. Xu, P. Athavale, A. Nachman, and G. A. Wright, "Multiscale registration of real-time and prior mri data for image-guided cardiac interventions," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 10, pp. 2621–2632, 2014.

[20] K. Mathiassen, J. E. Fjellin, K. Glette, P. K. Hol, and O. J. Elle, "An ultrasound robotic system using the commercial robot ur5," *Frontiers in Robotics and AI*, vol. 3, p. 1, 2016.

[21] R. Sznitman, S. Billings, D. Rother, D. Mirota, Y. Yang, J. Handa, P. Gehlbach, J. U. Kang, G. D. Hager, and R. Taylor, "Active multispectral illumination and image fusion for retinal microsurgery," in *International Conference on Information Processing in Computer-Assisted Interventions*, pp. 12–22, Springer, 2010.

[22] M. P. Bard, A. Amelink, V. N. Hegt, W. J. Graveland, H. J. Sterenborg, H. C. Hoogsteden, and J. G. Aerts, "Measurement of hypoxia-related parameters in bronchial mucosa by use of optical spectroscopy," *American Journal of Respiratory and Critical Care Medicine*, vol. 171, no. 10, pp. 1178–1184, 2005.

[23] B. S. Sorg, B. J. Moeller, O. Donovan, Y. Cao, and M. W. Dewhirst, "Hyperspectral imaging of hemoglobin saturation in tumor microvasculature and tumor hypoxia development," *Journal of Biomedical Optics*, vol. 10, no. 4, pp. 044004–044004, 2005.

[24] N. T. Clancy, D. Stoyanov, V. Sauvage, D. James, G.-Z. Yang, and D. S. Elson, "A triple endoscope system for alignment of multispectral images of moving tissue," in *Biomedical Optics*, p. BTuD27, Optical Society of America, 2010.

[25] D. W. Roberts, J. W. Strohbehn, J. F. Hatch, W. Murray, and H. Kettenberger, "A frameless stereotaxic integration of computerized tomographic imaging and the operating microscope," *Journal of Neurosurgery*, vol. 65, no. 4, pp. 545–549, 1986.

[26] Y. Enchev, "Neuronavigation: geneology, reality, and prospects," *Neurosurgical Focus*, vol. 27, no. 3, p. E11, 2009.

[27] M. Baumhauer, M. Feuerstein, H.-P. Meinzer, and J. Rassweiler, "Navigation in endoscopic soft tissue surgery: perspectives and limitations," *Journal of Endourology*, vol. 22, no. 4, pp. 751–766, 2008.

[28] D. Stoyanov, "Surgical vision," *Annals of Biomedical Engineering*, vol. 40, no. 2, pp. 332–345, 2012.

[29] S. DiMaio and C. Hasser, "The da vinci research interface," in *MICCAI Workshop on Systems and Arch. for Computer Assisted Interventions, Midas Journal*, 2008.

[30] J. Leven, D. Burschka, R. Kumar, G. Zhang, S. Blumenkranz, X. D. Dai, M. Awad, G. D. Hager, M. Marohn, M. Choti, *et al.*, "Davinci canvas: a telerobotic surgical system with integrated, robot-assisted, laparoscopic ultrasound capability," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2005 International Conference on*, pp. 811–818, Springer, 2005.

[31] M. Peterhans, A. vom Berg, B. Dagon, D. Inderbitzin, C. Baur, D. Candinas, and S. Weber, "A navigation system for open liver surgery: design, workflow and first clinical applications," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 7, no. 1, pp. 7–16, 2011.

[32] N. C. Buchs, F. Volonte, F. Pugin, C. Toso, M. Fusaglia, K. Gavaghan, P. E. Majno, M. Peterhans, S. Weber, and P. Morel, "Augmented environments for the targeting of hepatic lesions during image-guided robotic liver surgery," *Journal of Surgical Research*, vol. 184, no. 2, pp. 825–831, 2013.

[33] J. Ruurda, T. J. Van Vroonhoven, and I. Broeders, "Robot-assisted surgical systems: a new era in laparoscopic surgery.," *Annals of the Royal College of Surgeons of England*, vol. 84, no. 4, p. 223, 2002.

[34] G.-P. Haber, M. A. White, R. Autorino, P. F. Escobar, M. D. Kroh, S. Chalikonda, R. Khanna, S. Forest, B. Yang, F. Altunrende, *et al.*, "Novel robotic da vinci instruments for laparoendoscopic single-site surgery," *Urology*, vol. 76, no. 6, pp. 1279–1282, 2010.

[35] D. Kugelmann, L. Stratmann, N. Nühlen, F. Bork, S. Hoffmann, G. Samarbarksh, A. Pferschy, A. M. von der Heide, A. Eimannsberger, P. Fallavollita, *et al.*, "An augmented reality magic mirror as additive teaching device for gross anatomy," *Annals of Anatomy-Anatomischer Anzeiger*, vol. 215, pp. 71–77, 2018.

[36] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[37] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[38] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, pp. 809–817, 2013.

[39] H. Li, Y. Li, and F. Porikli, "Deeptrack: Learning discriminative feature representations online for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1834–1848, 2016.

[40] Z. Chi, H. Li, H. Lu, and M.-H. Yang, "Dual deep network for visual tracking," *arXiv preprint arXiv:1612.06053*, 2016.

[41] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, p. 58, 2013.

[42] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on*, pp. 2411–2418, IEEE, 2013.

[43] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, p. 13, 2006.

[44] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[45] B. Ristic, S. Arulampalam, and N. Gordon, "Beyond the kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 7, pp. 37–38, 2004.

[46] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[47] I. Biederman, "Recognition-by-components: a theory of human image understanding," *Psychological Review*, vol. 94, no. 2, p. 115, 1987.

[48] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[49] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[50] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[51] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003.

[52] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[53] D. Decarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *International Journal of Computer Vision*, vol. 38, no. 2, pp. 99–127, 2000.

[54] S. E. Palmer, *Vision Science: Photons to Phenomenology*. MIT press, 1999.

[55] V. Mahadevan and N. Vasconcelos, "Saliency-based discriminant tracking," in *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pp. 1007–1013, IEEE, 2009.

[56] H. Wu, G. Li, and X. Luo, "Weighted attentional blocks for probabilistic object tracking," *The Visual Computer*, vol. 30, no. 2, pp. 229–243, 2014.

[57] H. P. Moravec, "Rover visual obstacle avoidance," in *Artificial Intelligence (IJCAI), 1981 International Joint Conference on*, pp. 785–790, 1981.

[58] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999 IEEE International Conference on*, vol. 2, pp. 1150–1157, IEEE, 1999.

[59] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision (ECCV), 2006 European Conference on*, pp. 404–417, Springer, 2006.

[60] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE conference on*, pp. 510–517, IEEE, 2012.

[61] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," *Computer Vision (ECCV), 2010 European Conference on*, pp. 778–792, 2010.

[62] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2548–2555, IEEE, 2011.

[63] J. Pilet, V. Lepetit, and P. Fua, "Fast non-rigid surface detection, registration and realistic augmentation," *International Journal of Computer Vision*, vol. 76, no. 2, pp. 109–122, 2008.

[64] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[65] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision (ECCV), 2014 European Conference on*, pp. 818–833, Springer, 2014.

[66] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pp. 806–813, IEEE, 2014.

[67] C. Tomasi and T. Kanade, "Detection and tracking of point features," 1991.

[68] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Computer Vision (ECCV), 1998 European Conference on*, pp. 484–498, Springer, 1998.

[69] S. Sclaroff and J. Isidoro, "Active blobs," in *Computer Vision (ICCV), 1998 IEEE International Conference on*, pp. 1146–1153, IEEE, 1998.

[70] R. T. Collins, "Mean-shift blob tracking through scale space," in *Computer Vision and Pattern Recognition (CVPR), 2003 IEEE Conference on*, vol. 2, pp. II–234, IEEE, 2003.

[71] G. D. Hager, M. Dewan, and C. V. Stewart, "Multiple kernel tracking with ssd," in *Computer Vision and Pattern Recognition (CVPR), 2004 Conference on*, vol. 1, pp. I–I, IEEE, 2004.

[72] Y. Cheng, "Mean shift, mode seeking, and clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 8, pp. 790–799, 1995.

[73] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[74] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[75] X. Xie and K.-M. Lam, "Gabor-based kernel pca with doubly nonlinear mapping for face recognition with a single face image," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2481–2492, 2006.

[76] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1269–1276, IEEE, 2010.

[77] S. Avidan, "Ensemble tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 2, pp. 261–271, 2007.

[78] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pp. 983–990, IEEE, 2009.

[79] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *BMVC*, vol. 1, p. 6, 2006.

[80] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Computer Vision (ECCV), 2008 European Conference on*, pp. 234–247, Springer, 2008.

[81] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.

[82] S. Avidan, "Support vector tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 8, pp. 1064–1072, 2004.

[83] Y. Bai and M. Tang, "Robust tracking via weakly supervised ranking svm," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1854–1861, IEEE, 2012.

[84] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 3, pp. 583–596, 2015.

[85] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Computer Vision (ICCV), 2015 IEEE International Conference on*, pp. 3101–3109, IEEE, 2015.

[86] L. Deng, "Three classes of deep learning architectures and their applications: a tutorial survey," *APSIPA Transactions on Signal and Information Processing*, 2012.

[87] L. Deng, M. L. Seltzer, D. Yu, A. Acero, A.-r. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[88] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[89] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pp. 770–778, IEEE, 2016.

[90] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*, pp. 3431–3440, IEEE, 2015.

[91] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[92] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[93] R. Girshick, "Fast r-cnn," in *Computer Vision (ICCV), 2015 IEEE International Conference on*, pp. 1440–1448, IEEE, 2015.

[94] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[95] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision (ECCV), 2016 European Conference on*, pp. 21–37, Springer, 2016.

[96] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1653–1660, IEEE, 2014.

[97] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," in *Computer Vision (ECCV), 2016 European Conference on*, pp. 717–732, Springer, 2016.

[98] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1611.08050*, 2016.

[99] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015.

[100] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International Conference on Machine Learning*, pp. 597–606, 2015.

[101] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Computer Vision (ICCV), 2015 IEEE International Conference on*, pp. 3119–3127, IEEE, 2015.

[102] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," *arXiv preprint arXiv:1510.07945*, 2015.

[103] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1822–1829, IEEE, 2012.

[104] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 810–815, 2004.

[105] J. Xing, J. Gao, B. Li, W. Hu, and S. Yan, "Robust object tracking with online multi-lifespan dictionary learning," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 665–672, 2013.

[106] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 49–56, IEEE, 2010.

[107] Q. Yu, T. B. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Computer Vision (ECCV), 2008 European Conference on*, pp. 678–691, Springer, 2008.

[108] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 723–730, IEEE, 2010.

[109] T. Ortmaier, M. Gröger, and G. Hirzinger, "Motion estimation in minimally invasive beating heart surgery," in *International Journal of Computer Assisted Radiology and Surgery*, pp. 206–211, Springer, 2002.

[110] M. Gröger, T. Ortmaier, W. Sepp, and G. Hirzinger, "Tracking local motion on the beating heart," in *Medical Imaging*, pp. 233–241, International Society for Optics and Photonics, 2002.

[111] T. Ortmaier, M. Groger, D. H. Boehm, V. Falk, and G. Hirzinger, "Motion estimation in beating heart surgery," *Biomedical Engineering, IEEE Transactions on*, vol. 52, no. 10, pp. 1729–1740, 2005.

[112] D. Stoyanov, G. P. Mylonas, F. Deligianni, A. Darzi, and G. Z. Yang, "Soft-tissue motion tracking and structure estimation for robotic assisted mis procedures," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2005 International Conference on*, pp. 139–146, Springer, 2005.

[113] D. Stoyanov and G.-Z. Yang, "Stabilization of image motion for robotic assisted beating heart surgery," *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2007 International Conference on*, pp. 417–424, 2007.

[114] S. Giannarou, M. Visentini-Scarzanella, and G.-Z. Yang, "Affine-invariant anisotropic detector for soft tissue tracking in minimally invasive surgery," in *Biomedical Imaging (ISBI), 2009 IEEE International Symposium on*, pp. 1059–1062, IEEE, 2009.

[115] P. Mountney, B. Lo, S. Thiemjarus, D. Stoyanov, and G. Zhong-Yang, "A probabilistic framework for tracking deformable soft tissue in minimally invasive surgery," *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2007 International Conference on*, pp. 34–41, 2007.

[116] W. W. Lau, N. A. Ramey, J. J. Corso, N. V. Thakor, and G. D. Hager, "Stereo-based endoscopic tracking of cardiac surface deformation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2004 International Conference on*, pp. 494–501, Springer, 2004.

[117] D. Stoyanov, A. Darzi, and G. Z. Yang, "Dense 3d depth recovery for soft tissue deformation during robotically assisted laparoscopic surgery," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2004 International Conference on*, pp. 41–48, Springer, 2004.

[118] D. Stoyanov, A. Darzi, and G.-Z. Yang, "Laparoscope self-calibration for robotic assisted minimally invasive surgery," *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2005 International Conference on*, pp. 114–121, 2005.

[119] R. Richa, P. Poignet, and C. Liu, "Efficient 3d tracking for motion compensation in beating heart surgery," *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2008 International Conference on*, pp. 684–691, 2008.

[120] R. Richa, P. Poignet, and C. Liu, "Three-dimensional motion tracking for beating heart surgery using a thin-plate spline deformable model," *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 218–230, 2010.

[121] R. Richa, A. P. Bó, and P. Poignet, "Robust 3d visual tracking for robotic-assisted cardiac interventions," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2010 International Conference on*, pp. 267–274, Springer, 2010.

[122] R. Linhares, R. Richa, R. de Moraes, A. Sobieranski, and A. von Wangenheim, "Non-rigid fine adjustment of retina maps acquired using a slit-lamp," in *Computer-Based Medical Systems (CBMS), 2016 IEEE International Symposium on*, pp. 285–289, IEEE, 2016.

[123] S. Giannarou, M. Ye, G. Gras, K. Leibrandt, H. J. Marcus, and G.-Z. Yang, "Vision-based deformation recovery for intraoperative force estimation of tool–tissue interaction for neurosurgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 6, pp. 929–936, 2016.

[124] P. Mountney, D. Stoyanov, and G.-Z. Yang, "Three-dimensional tissue deformation recovery and tracking," *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 14–24, 2010.

[125] P. Kazanzides, Z. Chen, A. Deguet, G. Fischer, R. Taylor, and S. Dimaio, "An open-source research kit for the da vinci® surgical robot," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014.

[126] D. R. Uecker, Y. Wang, C. Lee, and Y. Wang, "Automated instrument tracking in robotically assisted laparoscopic surgery," *Journal of Image Guided Surgery*, vol. 1, no. 6, pp. 308–325, 1995.

[127] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Real-time visual servoing for laparoscopic surgery. controlling robot motion with color image segmentation," *IEEE Engineering in Medicine and Biology Magazine*, vol. 16, no. 1, pp. 40–45, 1997.

[128] O. Tonet, R. U. Thoranaghatte, G. Megali, and P. Dario, "Tracking endoscopic instruments without a localizer: a shape-analysis-based approach," *Computer Aided Surgery*, vol. 12, no. 1, pp. 35–42, 2007.

[129] X. Zhang and S. Payandeh, "Application of visual tracking for robot-assisted laparoscopic surgery," *Journal of Field Robotics*, vol. 19, no. 7, pp. 315–328, 2002.

[130] L. Zhang, M. Ye, P.-L. Chan, and G.-Z. Yang, "Real-time surgical tool tracking and pose estimation using a hybrid cylindrical marker," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 6, pp. 921–930, 2017.

[131] S. McKenna, H. N. Charif, and T. Frank, "Towards video understanding of laparoscopic surgery: Instrument tracking," in *Proceedings of Image and Vision Computing*, 2005.

[132] A. M. Cano, F. Gayá, P. Lamata, P. Sánchez-González, and E. J. Gómez, "Laparoscopic tool tracking method for augmented reality surgical applications," in *International Symposium on Biomedical Simulation*, pp. 191–196, Springer, 2008.

[133] A. Reiter and P. K. Allen, "An online learning approach to in-vivo tracking using synergistic features," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 3441–3446, IEEE, 2010.

[134] Z. Pezzementi, S. Voros, and G. D. Hager, "Articulated object tracking by rendering consistent appearance parts," in *Robotics and Automation (ICRA), 2009 IEEE International Conference on*, pp. 3940–3947, IEEE, 2009.

[135] M. Allan, S. Ourselin, S. Thompson, D. J. Hawkes, J. Kelly, and D. Stoyanov, "Toward detection and localization of instruments in minimally invasive surgery," *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 4, pp. 1050–1058, 2013.

[136] M. Allan, S. Thompson, M. J. Clarkson, S. Ourselin, D. J. Hawkes, J. Kelly, and D. Stoyanov, "2d-3d pose tracking of rigid instruments in minimally invasive surgery," in *Information Processing in Computer-assisted Interventions, International Conference on*, pp. 1–10, Springer, 2014.

[137] M. Allan, P.-L. Chang, S. Ourselin, D. J. Hawkes, A. Sridhar, J. Kelly, and D. Stoyanov, "Image based surgical instrument pose estimation with multi-class labelling and optical flow," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015 International Conference on*, pp. 331–338, Springer, 2015.

[138] M. Alsheakhali, M. Yigitsoy, A. Eslami, and N. Navab, "Real time medical instrument detection and tracking in microsurgery," in *Bildverarbeitung für die Medizin*, pp. 185–190, Springer, 2015.

[139] M. Ye, L. Zhang, S. Giannarou, and G.-Z. Yang, "Real-time 3d tracking of articulated tools for robotic surgery," in *edical Image Computing and Computer-Assisted Intervention (MICCAI), 2016 International Conference on,*, pp. 386–394, Springer, 2016.

[140] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "Endonet: A deep architecture for recognition tasks on laparoscopic videos," *Medical Imaging, IEEE Transactions on*, vol. 36, no. 1, pp. 86–97, 2017.

[141] M. Sahu, A. Mukhopadhyay, A. Szengel, and S. Zachow, "Addressing multi-label imbalance problem of surgical tool detection using cnn," *International Journal of Computer Assisted Radiology and Surgery*, pp. 1–8, 2017.

[142] K. Mishra, R. Sathish, and D. Sheet, "Learning latent temporal connectionism of deep residual visual abstractions for identifying surgical tools in laparoscopy procedures," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pp. 2233–2240, IEEE, 2017.

[143] Z. Zhao, S. Voros, Y. Weng, F. Chang, and R. Li, "Tracking-by-detection of surgical instruments in minimally invasive surgery via the convolutional neural network deep learning-based method," *Computer Assisted Surgery*, pp. 1–10, 2017.

[144] B. Choi, K. Jo, S. Choi, and J. Choi, "Surgical-tools detection based on convolutional neural network in laparoscopic robot-assisted surgery," in *Engineering in Medicine and Biology Society (EMBC), 2017 IEEE International Conference on*, pp. 1756–1759, IEEE, 2017.

[145] R. Richa, M. Balicki, E. Meisner, R. Sznitman, R. Taylor, and G. Hager, "Visual tracking of surgical tools for proximity detection in retinal surgery," *Information Processing in Computer-Assisted Interventions*, pp. 55–66, 2011.

[146] R. Sznitman, A. Basu, R. Richa, J. Handa, P. Gehlbach, R. H. Taylor, B. Jedynak, and G. D. Hager, "Unified detection and tracking in retinal microsurgery," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2011 International Conference on*, pp. 1–8, Springer, 2011.

[147] R. Sznitman, R. Richa, R. H. Taylor, B. Jedynak, and G. D. Hager, "Unified detection and tracking of instruments during retinal microsurgery," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 5, pp. 1263–1273, 2013.

[148] R. Sznitman, K. Ali, R. Richa, R. Taylor, G. Hager, and P. Fua, "Data-driven visual tracking in retinal microsurgery," *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2012 International Conference on*, pp. 568–575, 2012.

[149] N. Rieke, D. J. Tan, M. Alsheakhali, F. Tombari, C. A. di San Filippo, V. Belagiannis, A. Eslami, and N. Navab, "Surgical tool tracking and pose estimation in retinal microsurgery," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015 International Conference on*, pp. 266–273, Springer, 2015.

[150] N. Rieke, D. J. Tan, C. A. di San Filippo, F. Tombari, M. Alsheakhali, V. Belagiannis, A. Eslami, and N. Navab, "Real-time localization of articulated surgical instruments in retinal microsurgery," *Medical Image Analysis*, vol. 34, pp. 82–100, 2016.

[151] J. P. Vizcaíno, N. Rieke, D. J. Tan, F. Tombari, A. Eslami, and N. Navab, "Automatic initialization and failure detection for surgical tool tracking in retinal microsurgery," in *Bildverarbeitung für die Medizin 2017*, pp. 346–351, Springer, 2017.

[152] M. Alsheakhali, A. Eslami, and N. Navab, "Detection of articulated instruments in retinal microsurgery," in *Biomedical Imaging (ISBI), 2016 IEEE International Symposium on*, pp. 107–110, IEEE, 2016.

[153] T. Probst, K.-K. Maninis, A. Chhatkuli, M. Ourak, E. V. Poorten, and L. Van Gool, "Automatic tool landmark detection for stereo vision in robot-assisted retinal surgery," *arXiv preprint arXiv:1709.05665*, 2017.

[154] I. Laina, N. Rieke, C. Rupprecht, J. P. Vizcaíno, A. Eslami, F. Tombari, and N. Navab, "Concurrent segmentation and localization for tracking of surgical instruments," *arXiv preprint arXiv:1703.10701*, 2017.

[155] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M. M. A. Sanchez, and J. Marescaux, "Active filtering of physiological motion in robotized surgery using predictive control," *Robotics, IEEE Transactions on*, vol. 21, no. 1, pp. 67–79, 2005.

[156] D. Stoyanov, A. Rayshubskiy, and E. Hillman, "Robust registration of multispectral images of the cortical surface in neurosurgery," in *Biomedical Imaging (ISBI), 2012 IEEE International Symposium on*, pp. 1643–1646, IEEE, 2012.

[157] P. Mountney and G.-Z. Yang, "Soft tissue tracking for minimally invasive surgery: Learning local deformation online," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2008 International Conference on*, pp. 364–372, Springer, 2008.

[158] T. J. Ortmaier, *Motion compensation in minimally invasive robotic surgery*. PhD thesis, Universität München, 2003.

[159] D. Stoyanov, A. Darzi, and G. Z. Yang, "A practical approach towards accurate dense 3d depth recovery for robotic laparoscopic surgery," *Computer Aided Surgery*, vol. 10, no. 4, pp. 199–208, 2005.

[160] S. Giannarou, M. Visentini-Scarzanella, and G.-Z. Yang, "Probabilistic tracking of affine-invariant anisotropic regions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 130–143, 2013.

[161] G. A. Puerto-Souza and G. L. Mariottini, "Hierarchical multi-affine (hma) algorithm for fast and accurate feature matching in minimally-invasive surgical images," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2007–2012, IEEE, 2012.

[162] M. C. Yip, D. G. Lowe, S. E. Salcudean, R. N. Rohling, and C. Y. Nguan, "Real-time methods for long-term tissue feature tracking in endoscopic scenes," in *Information Processing in Computer-Assisted Interventions*, pp. 33–43, Springer, 2012.

[163] J. Braux-Zin, R. Dupont, and A. Bartoli, "Combining features and intensity for wide-baseline non-rigid surface registration," in *British Machine Vision Conference (BMVC)*, BMVA Press, 2013.

[164] L. Maier-Hein, P. Mountney, A. Bartoli, H. Elhawary, D. Elson, A. Groch, A. Kolb, M. Rodrigues, J. Sorger, S. Speidel, and D. Stoyanov, "Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery," *Medical Image Analysis*, vol. 17, no. 8, pp. 974–996, 2013.

[165] G. Bradski, "The opencv library," *Doctor Dobbs Journal*, vol. 25, no. 11, pp. 120–126, 2000.

[166] P. Fua and C. Brechbühler, "Imposing hard constraints on soft snakes," in *Computer Vision (ECCV), 1996 European Conference on*, pp. 495–506, Springer, 1996.

[167] J. Zhu, M. R. Lyu, and T. S. Huang, "A fast 2d shape recovery approach by fusing features and appearance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 7, pp. 1210–1224, 2009.

[168] S. S. Keerthi and D. DeCoste, "A modified finite newton method for fast solution of large scale linear svms," in *Journal of Machine Learning Research*, pp. 341–361, 2005.

[169] O. L. Mangasarian, "A finite newton method for classification," *Optimization Methods and Software*, vol. 17, no. 5, pp. 913–929, 2002.

[170] M. R. Pickering, A. A. Muhit, J. M. Scarvell, and P. N. Smith, "A new multi-modal similarity measure for fast gradient-based 2d-3d image registration," in *Engineering in Medicine and Biology Society (EMBC), 2009 IEEE International Conference on*, pp. 5821–5824, IEEE, 2009.

[171] R. Richa, R. Sznitman, R. Taylor, and G. Hager, "Visual tracking using the sum of conditional variance," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference On*, pp. 2953–2958, IEEE, 2011.

[172] M. Gröger, W. Sepp, T. Ortmaier, and G. Hirzinger, "Reconstruction of image structure in presence of specular reflections," in *Joint Pattern Recognition Symposium*, pp. 53–60, Springer, 2001.

[173] D. Stoyanov and G.-Z. Yang, "Removing specular reflection components for robotic assisted laparoscopic surgery," in *Image Processing (ICIP), 2005 IEEE International Conference on*, vol. 3, pp. III–632, IEEE, 2005.

[174] D. Stoyanov, "Stereoscopic scene flow for robotic assisted minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2012 International Conference on*, pp. 479–486, Springer, 2012.

[175] F. Selka, S. A. Nicolau, V. Agnus, A. Bessaid, J. Marescaux, and L. Soler, "Evaluation of endoscopic image enhancement for feature tracking: A new validation framework," in *Augmented Reality Environments for Medical Imaging and Computer-Assisted Interventions*, pp. 75–85, Springer, 2013.

[176] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Pattern Recognition (ICPR), 2010 International Conference on*, pp. 2756–2759, IEEE, 2010.

[177] N. T. Clancy, D. Stoyanov, D. R. James, A. Di Marco, V. Sauvage, J. Clark, G.-Z. Yang, and D. S. Elson, "Multispectral image alignment using a three channel endoscope in vivo during minimally invasive surgery," *Biomedical Optics Express*, vol. 3, no. 10, pp. 2567–2578, 2012.

[178] B. Delabarre and E. Marchand, "Visual servoing using the sum of conditional variance," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 1689–1694, IEEE, 2012.

[179] N. T. Clancy, S. Saso, D. Stoyanov, V. Sauvage, D. J. Corless, M. Boyd, D. E. Noakes, M.-Y. Thum, S. Ghaem-Maghami, J. R. Smith, *et al.*, "Multispectral imaging of organ viability during uterine transplantation surgery," in *Advanced Biomedical and Clinical Diagnostic Systems XII*, vol. 8935, p. 893510, International Society for Optics and Photonics, 2014.

[180] N. T. Clancy, S. Arya, D. Stoyanov, M. Singh, G. B. Hanna, and D. S. Elson, "Intraoperative measurement of bowel oxygen saturation using a multispectral imaging laparoscope," *Biomedical Optics Express*, vol. 6, no. 10, pp. 4179–4190, 2015.

[181] S. J. Wirkert, N. T. Clancy, D. Stoyanov, S. Arya, G. B. Hanna, H.-P. Schlemmer, P. Sauer, D. S. Elson, and L. Maier-Hein, "Endoscopic sheffield index for unsupervised in vivo spectral band selection," in *International Workshop on Computer-Assisted and Robotic Endoscopy*, pp. 110–120, Springer, 2014.

[182] G. Jones, N. T. Clancy, X. Du, M. Robu, S. Arridge, D. S. Elson, and D. Stoyanov, "Fast estimation of haemoglobin concentration in tissue via wavelet decomposition," in

Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2017 International Conference on, pp. 100–108, Springer, 2017.

[183] T. Pock, M. Unger, D. Cremers, and H. Bischof, "Fast and exact solution of total variation models on the gpu," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pp. 1–8, IEEE, 2008.

[184] A. Reiter, P. K. Allen, and T. Zhao, "Learning features on robotic surgical tools," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Conference on*, pp. 38–43, IEEE, 2012.

[185] J. Ren, R. V. Patel, K. A. McIsaac, G. Guiraudon, and T. M. Peters, "Dynamic 3d virtual fixtures for minimally invasive beating heart procedures.," *Medical Imaging, IEEE Transactions on*, vol. 27, no. 8, pp. 1061–1070, 2008.

[186] A. M. Okamura, "Haptic feedback in robot-assisted minimally invasive surgery," *Current Opinion in Urology*, vol. 19, no. 1, p. 102, 2009.

[187] L. Joskowicz, C. Milgrom, A. Simkin, L. Tockus, and Z. Yaniv, "Fracas: A system for computer-aided image-guided long bone fracture surgery," *Computer Aided Surgery*, vol. 3, no. 6, pp. 271–288, 1998.

[188] A. Reiter, P. K. Allen, and T. Zhao, "Feature classification for tracking articulated surgical tools," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2012 International Conference on*, pp. 592–600, Springer, 2012.

[189] S. Speidel, A. Kroehnert, S. Bodenstedt, H. Kenngott, B. Mueller-Stich, and R. Dillmann, "Image-based tracking of the suturing needle during laparoscopic interventions," in *SPIE Medical Imaging*, pp. 94150B–94150B, International Society for Optics and Photonics, 2015.

[190] R. Sznitman, C. Becker, and P. Fua, "Fast part-based classification for instrument detection in minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2014 International Conference on*, pp. 692–699, Springer, 2014.

[191] S. Speidel, E. Kuhn, S. Bodenstedt, S. Röhl, H. Kenngott, B. Müller-Stich, and R. Dillmann, "Visual tracking of da vinci instruments for laparoscopic surgery," in *Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 9036, p. 903608, International Society for Optics and Photonics, 2014.

[192] D. Bouget, R. Benenson, M. Omran, L. Riffaud, B. Schiele, and P. Jannin, "Detecting surgical tools by modelling local appearance and global shape," *Medical Imaging, IEEE Transactions on*, vol. 34, no. 12, pp. 2603–2617, 2015.

[193] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[194] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[195] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1631–1643, 2005.

[196] S. Duffner and C. Garcia, "Pixeltrack: A fast adaptive algorithm for tracking non-rigid objects," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 2480–2487, IEEE, 2013.

[197] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," in *Journal of Machine Learning Research*, pp. 1453–1484, 2005.

[198] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Computer Vision (ECCV), 2014 European Conference on*, pp. 188–203, Springer, 2014.

[199] D. Chen, Z. Yuan, Y. Wu, G. Zhang, and N. Zheng, "Constructing adaptive complex cells for robust visual tracking," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 1113–1120, IEEE, 2013.

[200] L. Zhang and L. J. van der Maaten, "Preserving structure in model-free tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 4, pp. 756–769, 2014.

[201] S. He, Q. Yang, R. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2427–2434, IEEE, 2013.

[202] D.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "Visual tracking using pertinent patch selection and masking," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 3486–3493, IEEE, 2014.

[203] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1245–1256, 2013.

[204] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pp. 175–181, Morgan Kaufmann Publishers Inc., 1997.

[205] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.

[206] J. Shi *et al.*, "Good features to track," in *Computer Vision and Pattern Recognition (CVPR), 1994 IEEE Conference on*, pp. 593–600, IEEE, 1994.

[207] X. Li, C. Shen, A. Dick, and A. Hengel, "Learning compact binary codes for visual tracking," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2419–2426, IEEE, 2013.

[208] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1177–1184, IEEE, 2011.

[209] J. Kwon and K. M. Lee, "Tracking by sampling and integratingmultiple trackers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 7, pp. 1428–1441, 2014.

[210] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Computer Vision (ECCV), 2014 European Conference on*, pp. 188–203, Springer, 2014.

[211] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference (BMVC)*, BMVA Press, 2014.

[212] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Computer Vision (ECCV), 2014 European Conference on*, pp. 254–265, Springer, 2014.

[213] D.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "Multihypothesis trajectory analysis for robust visual tracking," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 5088–5096, IEEE, 2015.

[214] D. Chen, Z. Yuan, G. Hua, Y. Wu, and N. Zheng, "Description-discrimination collaborative tracking," in *Computer Vision (ECCV), 2014 European Conference on*, pp. 345–360, Springer, 2014.

[215] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Computer Vision (ECCV), 2002 European Conference on*, pp. 661–675, Springer, 2002.

[216] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1910–1917, IEEE, 2012.

[217] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1830–1837, IEEE, 2012.

[218] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1940–1947, IEEE, 2012.

[219] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1313–1320, IEEE, 2011.

[220] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2042–2049, IEEE, 2012.

[221] J. P. Lewis, "Fast template matching," in *Vision Interface*, vol. 95, pp. 15–19, 1995.

[222] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1195–1202, IEEE, 2011.

[223] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay, T. Vojír, G. Fernández, and A. Lukezic, "The visual object tracking vot2014 challenge results," in *Computer Vision Workshops (ECCVW), 2015 European Conference on*, pp. 191–217, 2015.

[224] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernández, T. Vojir, G. Hager, G. Nebehay, and et. al, "The visual object tracking vot2015 challenge results," in *Computer Vision (ICCV), 2015 IEEE International Conference Workshops on*, Dec 2015.

[225] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Computer Vision Workshops (ICCVW), 2015 IEEE International Conference on*, pp. 58–66, IEEE, 2015.

[226] N. Wang and D.-Y. Yeung, "Ensemble-based tracking: Aggregating crowdsourced structured time series data," in *Machine Learning, International Conference on*, pp. 1107–1115, 2014.

[227] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Computer Vision (ICCV), 2015 IEEE International Conference on*, pp. 4310–4318, IEEE, 2015.

[228] A. Lukežič, L. Čehovin, and M. Kristan, "Deformable parts correlation filters for robust visual tracking," *arXiv preprint arXiv:1605.03720*, 2016.

[229] Y. Hua, K. Alahari, and C. Schmid, "Online object tracking with proposal selection," in *Computer Vision (ICCV), 2015 IEEE International Conference on*, pp. 3092–3100, IEEE, 2015.

[230] M. Zhang, J. Xing, J. Gao, X. Shi, Q. Wang, and W. Hu, "Joint scale-spatial correlation tracking with adaptive rotation estimation," in *omputer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pp. 32–40, IEEE, 2015.

[231] R. Sznitman, C. Becker, and P. Fua, "Fast part-based classification for instrument detection in minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2014 International Conference on*, pp. 692–699, Springer, 2014.

[232] X. Du, M. Allan, A. Dore, S. Ourselin, D. Hawkes, J. D. Kelly, and D. Stoyanov, "Combined 2d and 3d tracking of surgical instruments for minimally invasive and robotic-assisted surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 6, pp. 1109–1119, 2016.

[233] A. Reiter, P. K. Allen, and T. Zhao, "Articulated surgical tool detection using virtually-rendered templates," in *International Journal of Computer Assisted Radiology and Surgery*, 2012.

[234] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[235] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[236] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015 International Conference on*, pp. 234–241, Springer, 2015.

[237] L. C. García-Peraza-Herrera, W. Li, C. Gruijthuijsen, A. Devreker, G. Attilakos, J. Deprest, E. Vander Poorten, D. Stoyanov, T. Vercauteren, and S. Ourselin, "Real-time segmentation of non-rigid surgical tools based on deep learning and tracking," in *International Workshop on Computer-Assisted and Robotic Endoscopy*, pp. 84–95, Springer, 2016.

[238] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, pp. 2843–2851, 2012.

[239] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Pattern Recognition (ICPR), 2006 International Conference on*, vol. 3, pp. 850–855, IEEE, 2006.

[240] J. Schwartz, A. Steger, and A. Weißl, "Fast algorithms for weighted bipartite matching," in *Experimental and Efficient Algorithms, International Workshop on*, pp. 476–487, Springer, 2005.

[241] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics (NRL)*, vol. 2, no. 1-2, pp. 83–97, 1955.

[242] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pp. 1–8, IEEE, 2008.

[243] T. Kurmann, P. M. Neila, X. Du, P. Fua, D. Stoyanov, S. Wolf, and R. Sznitman, "Simultaneous recognition and pose estimation of instruments in minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2017 International Conference on*, pp. 505–513, Springer, 2017.